UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Contribuições ao Estudo da Dinâmica na Teoria da Informação: Aplicações em Clustering Dinâmico

Amanda Gondim de Oliveira

Orientador: Prof. Dr. Adrião Duarte Dória Neto

Co-orientador: Prof. Dr. Allan Medeiros de Martins

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Doutor em Ciências.

Número de ordem PPgEEC: D225
Natal, RN, julho de 2018

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Oliveira, Amanda Gondim de.
Contribuições ao estudo da dinâmica na teoria da informação: aplicações em clustering dinâmico / Amanda Gondim de Oliveira. - 2018.
94 f.: il.

Tese (doutorado) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e de Computação. Natal, RN, 2018.
Orientador: Prof. Dr. Adrião Duarte Dória Neto.
Coorientador: Prof. Dr. Allan Medeiros de Martins.

1. Teoria da probabilidade - Tese. 2. Clustering dinâmico - Tese. 3. Teoria da informação - Tese. 4. Processos dinâmicos - Tese. I. Dória Neto, Adrião Duarte. II. Martins, Allan Medeiros de. III. Título.

RN/UF/BCZM CDU 519.21(043.2)

Elaborado por FERNANDA DE MEDEIROS FERREIRA AQUINO - CRB-15/301

Universidade Federal do Rio Grande do Norte

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

ATA Nº 225

ATA Nº 225

Aos três dias do mês de agosto do ano de dois mil e dezoito, foi realizada a 225ª sessão de defesa de tese de doutorado do Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN, na qual o doutorando Amanda Gondim de Oliveira apresentou o trabalho que tem como título: Contribuições ao Estudo da Dinâmica na Teoria da Informação: Aplicações em Clustering Dinâmico. A sessão teve início às 09h00min, tendo a banca examinadora sido constituída pelos seguintes participantes: Adriao Duarte Doria Neto (Dr. UFRN, Presidente), Aluisio Igor Rego Fontes (Dr. IFRN, Examinador Externo à Instituição), Guilherme de Alencar Barreto (Dr. UFC, Examinador Externo à Instituição), Allan de Medeiros Martins (Dr. UFRN, Examinador Interno ao Programa) e Daniel Sabino Amorim de Araujo (Dr. UFRN, Examinador Externo ao Programa). Após a apresentação do trabalho e o exame pela banca, o doutorando foi considerado APROVADA, tendo sido lavrada a presente ata, que vai assinada pelos examinadores e pelo doutorando. A versão final da tese deverá ser entregue ao programa, no prazo máximo de 60 dias, contendo as modificações sugeridas pela banca examinadora e constante na folha de correção anexa. Conforme o Artigo 49 da Resolução 197/2013 - CONSEPE, o candidato não terá o título se não cumprir as exigências acima.

Aluisio Igor R. Fontes

Dr. ALUISIO IGOR REGO FONTES, IFRN

Examinador Externo à Instituição

Guilherme de Alencar Barreto

Dr. GUILHERME DE ALENCAR BARRETO, UFC

Examinador Externo à Instituição

Dr. DANIEL SABINO AMORIM DE ARAUJO, UFRN

Examinador Externo ao Programa

Dr. ALLAN DE MEDEIROS MARTINS, UFRN

Examinador Interno

Adrião Duarte Doria Neto

Dr. ADRIAO DUARTE DORIA NETO, UFRN

Presidente

AMANDA GONDIM DE OLIVEIRA

Agradecimentos

Meus agradecimentos vão em primeiro lugar, aos meus orientadores Prof. Dr. Adrião Duarte Dória Neto e ao Prof Dr. Allan de Medeiros Martins que me acolheram como orientanda e me ajudaram durante o desenvolvimento desse trabalho.

Aos colegas da UFERSA pelo apoio durante meu período de afastamento nas atividades acadêmicas na instituição.

À minha família, em especial, meu marido Felipe, meu filho Lucas, meus pais Samuel e Nalva, e minha irmã Luana, pelo apoio durante esta jornada.

Aos amigos do Laboratório de Sistemas Inteligentes pelas observações e sugestões dadas ao longo desses anos.

Ao PPGEEC e UFRN pelo apoio acadêmico e financeiro obtido na divulgação dos trabalhos científicos.

Resumo

A Teoria da Informação é um ramo da matemática, mais especificamente da teoria de probabilidades, que estuda quantificação da informação. Recentemente, várias pesquisas tem tido sucesso com o uso do aprendizado de máquina por teoria da informação (ITL) como uma nova técnica de aprendizado não supervisionado, onde as medidas de informação são usadas como critério de otimalidade no aprendizado. Neste trabalho iremos analisar um aspecto ainda não explorado destas medidas de informação, o seu comportamento dinâmico. O principal objetivo desse trabalho é investigar o uso de medidas da teoria da informação no contexto dos processos dinâmicos. Para isso, o mesmo foi realizado em 3 (três) fases distintas. O objetivo na primeira fase desse trabalho foi investigar a presença de dinâmica na informação de processos que evoluem com o tempo. Como fonte de informação dinâmica, foram utilizados vídeos com diferentes características. O objetivo da segunda fase foi apresentar uma nova representação para processos dinâmicos em espaço de estados chamado Modelo Estados de Informação (MEI). Nesta representação, os estados do sistema são descritos em função das medidas de informação dos mesmos. Para validar esta nova forma de representação, foram realizados alguns experimentos com vídeos com o objetivo de avaliar a qualidade dos mesmos quando submetidos a diferentes aspectos dinâmicos. Na terceira fase, investigou-se o uso das medidas baseadas na teoria da informação dentro da área de clustering dinâmico. O objetivo nessa fase foi comparar o desempenho do uso das medidas da teoria da informação com as medidas tradicionais nas operações de fusão e separação entre clusters. Os resultados obtidos em todas as fases foram bastante satisfatórios atendendo os objetivos propostos no trabalho.

Palavras-chave: Teoria da Informação, Clustering Dinâmico, Processos Dinâmicos

Abstract

Information Theory is a branch of mathematics, more specifically probability theory, that studies information quantification. Recently, several researches have been successful with the use of Information Theoretic Learning (ITL) as a new technique of unsupervised learning. In these works, information measures are used as criterion of optimality in learning. In this work, we will analyze a still unexplored aspect of these information measures, their dynamic behavior. The main objective of this work is to investigate the use of measures of information theory in the context of dynamic processes. For this, the same was done in 3 (three) distinct phases. In the first phase we investigated the presence of dynamics in the information in the processes. As a source of dynamic information, videos with different characteristics were used. The second phase presents a new representation for dynamical processes by state space called Information State Representation. In this representation, the states of the system are described as a function of the information measures of the system. To validate this new form of representation, some experiments were carried out with videos aiming at evaluating its quality when submitted to different dynamic aspects. In the third phase, we investigated the use of measures based on information theory within the area of dynamic clustering. The objective in this phase was to compare the performance of the use of measures of information theory with traditional measurements in the operations of merge and split between clusters. The results obtained in all the phases were quite satisfactory meeting the objectives proposed in the work.

Keywords: Information Theory, Dynamic Clustering, Dynamic Processes.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Recentes pesquisas (Liu et al. 2006, Santamaria et al. 2006) vem estudando conceitos de aprendizado por teoria da informação (ITL) para estender conceitos úteis de técnicas estatísticas lineares e de segunda ordem para aprendizagem estatística de ordem superior e não-linear. A pesquisa com ITL iniciou-se nos anos 90 e o seu conceito básico consiste no uso de medidas da teoria da informação estimados diretamente a partir dos dados para substituir os descritores estatísticos convencionais de média, variância e covariância. Para isto, é necessária a medição de grandezas como entropia e informação mútua a partir de amostras, sem o conhecimento das distribuições de probabilidade que geraram tais amostras. Dessa forma, as duas grandes ferramentas propostas na área de ITL são o potencial de informação e a correntropia. Estas grandezas quantificam de maneira indireta a entropia e a entropia cruzada de uma fonte de dados utilizando diretamente amostras sem o conhecimento a priori de sua distribuição.

1.1 Motivação

Apesar de ser uma área em expansão, a grande maioria das pesquisas publicadas na área de ITL envolvem apenas dados estáticos (Gokcay & Principe 2002) (Martins et al. 2014) (Araújo 2013), cujas propriedades estatísticas não mudam ao longo do tempo, o que não representa a grande maioria dos sistemas reais que são dinâmicos e quase sempre ruidosos.

A possibilidade de se estimar grandezas estatísticas, como entropia, a partir das amostras, propicia o estudo de ambientes onde se tenha processos estocásticos. Esse fato se mostra bastante interessante quando lembramos que existe uma forte interação entre a teoria que estuda sistemas dinâmicos e a que estuda probabilidades e modelos estocásticos, já que a grande maioria dos sistemas reais são dinâmicos e estão sujeitos a ruídos. Desse modo, os sistemas dinâmicos estocásticos vem ganhando cada vez mais interesse nas últimas décadas pela diversidade de problemas cuja modelagem inclui algum aspecto probabilístico.

Um exemplo bastante comum de ambiente dinâmico são os vídeos. Em cada frame de um vídeo, existem informações de cores presentes nas imagens que o compõem. Desse modo, é possível considerar cada pixel, de cada frame, como uma amostra de uma variável aleatória que evolui no tempo.

Utilizando as técnicas presentes em ITL, é possível estimar grandezas como entropia a cada instante de tempo e estudar sua evolução. Além da entropia, é possível definir outras estimativas de informação, como a informação mútua entre blocos de dados em um mesmo instante de tempo. Diante disso, podemos pensar em definir uma série temporal que envolva essas medidas de informação, o que consequentemente nos traz alguns questionamentos: Estas estimativas possuem dinâmica? Existe alguma relação entre as estimativas e seus valores passados? Existe alguma estrutura temporal?

1.2 Objetivos e Contribuições

O principal objetivo desse trabalho é investigar o uso de descritores da teoria da informação dentro do contexto dos processos dinâmicos. Para alcançar esse objetivo, o trabalho foi realizado em fases, onde os aspectos teóricos foram apresentados e algumas aplicações realizadas com o intuito de ilustrar os conceitos.

O primeiro objetivo desse trabalho foi investigar a presença de dinâmica na informação de processos que evoluem com o tempo. Para isso, fizemos uso de vídeos como exemplos de processos dinâmicos reais. Os resultados dessa primeira fase da pesquisa podem ser visualizados no artigo "An Analysis of Information Dynamic Behavior"publicado na revista Entropy em novembro de 2017. Nessa primeira fase do trabalho, realizou-se uma análise do comportamento dinâmico das informações por meio do uso de uma rede neural autorregressiva (NAR), a qual foi capaz de identificar a existência de dinâmica na informação dos dados de forma adequada.

Com o intuito de expandir ainda mais a ideia de informação dinâmica, a segunda fase do trabalho consistiu em definirmos um modelo em espaço de estados baseado em informação dinâmica. Para isso, consideramos que as estimativas das medidas de informação formam um vetor de estados, que neste trabalho definiremos como Estados de Informação. O modelo em espaço de estados deve reger a evolução destes Estados de Informação.

Em outras palavras, neste trabalho definimos uma modelagem de processos dinâmicos baseada apenas na informação contida em seus dados. Para isso, sugerimos uma representação em espaço de estados, onde as variáveis de estado do sistema são modeladas como Estados de Informação.

Para ilustrar uma aplicação deste conceito, utilizamos como fonte de dados, frames de vídeos e implementamos um sistema capaz de inferir a qualidade da compressão de um vídeo sem que o sistema tenha acesso ao vídeo original sem compressão. Ou seja, dados dois vídeos comprimidos com taxas diferentes, o sistema é capaz de informar qual deles tem melhor qualidade de imagem. Esta aplicação tem uso em sistemas automáticos de separação e classificação de vídeo de grandes provedores como YouTube, Vimeo, entre outros. Os resultados dessa segunda fase da pesquisa podem ser visualizados no artigo "Information State: A Representation for Dynamic Processes Using Information Theory"publicado nos anais do IJCNN 2018.

A terceira fase desse trabalho consistiu em aplicar os conceitos de informação dinâmica na área de clustering dinâmico, a qual se encontra sendo bastante estudada atualmente em virtude de suas inúmeras possibilidades de aplicação.

Nas aplicações de hoje em dia, os stream de dados em evolução são onipresentes. De fato, exemplos de aplicações relevantes para streams de dados estão se tornando mais numerosos e mais importantes, incluindo detecção de intrusão em redes, fluxos de transações, registros de telefone, fluxos sociais e monitoramento do tempo.

Evolving cluster models (ECM) são projetados para explorar uma enorme quantidade de streams de dados on line (Bifet et al. 2010) ou bancos de dados muito grandes (VLDBs) por meio do agrupamento e da compressão de dados de uma forma rápida e incremental dentro de um contexto de aprendizado não supervisionado. Os ECMs também são referenciados como métodos de "dynamic clustering", “incremental clustering” ou “evolving clustering” (Bouchachia 2011), porque eles podem processar etapas de dados, atualizar e evoluir partições de cluster em etapas de aprendizado incremental.

Um bom algoritmo para clustering dinâmico deve considerar as operações de junção e separação entre os clusters após a chegada de novos stream de dados. Os trabalhos recentes fazem uso das mais diversas técnicas para esse fim, entretanto quase sempre essas técnicas estão relacionadas com as medidas de proximidades tradicionais. O objetivo dessa fase do trabalho foi o de investigar o uso de descritores da teoria da informação em substituição às medidas tradicionais nas operações de junção e separação desses algoritmos para clustering dinâmico. A grande vantagem do uso de medidas baseadas na teoria da informação é o fato das mesmas não fazerem suposições acerca dos dados, já que essas medidas são calculadas em função das próprias amostras. Esse fato, pode ser bastante útil já que um bom algoritmo deve ser o mais robusto possível à diferentes conjuntos de dados.

1.3 Estado da Arte

Por não fazer suposições sobre a natureza dos dados, o aprendizado por teoria da informação tem ganhado espaço no processo de aprendizagem. Diante disso, uma revisão da literatura de diversos trabalhos envolvendo ITL no contexto de clustering são apresentados nessa seção.

Em (Hofmann & Buhmann 1997),os autores utilizaram otimização combinatória em conjunto com o princípio da entropia máxima para agrupar dados. O trabalho de (Gokcay & Principe 2002) utilizou uma medida de similaridade baseada na entropia de Rényi para estimar o custo de particionar um conjunto de dados. Ambos os trabalhos apresentaram resultados positivos superando as abordagens tradicionais.

Em (Martins 2005), propôs um método de clusterização baseado em medidas de dissimilaridade não-euclidianas que incorporam alguma informação sobre a estatística dos dados. O seu trabalho consistiu em quantizar vetorialmente com redes neurais competitivas todo o conjunto de dados em classes auxiliares modelando cada um delas como uma gaussiana. Uma vez modelados os conjuntos iniciais, os mesmos são ligados segundo um critério a fim de formar os clusters finais. O critério utilizado na ligação entre as classes auxiliares é feito com base no cálculo de uma medida de dissimilaridade entre cada par de classes. Entre as medidas de dissimilaridade utilizadas estão: distância de mahalanobis, distância de Bhattacharyya, divergênciade Kullback-Leibler, medidas de negentropia. Cada cluster formado é modelado então como uma mistura finita de gaussianas.

Outros trabalhos como (Rao et al. 2009) e (Araújo et al. 2013), utilizaram o Potencial de Informação Cruzado para realizar o agrupamento. Em 2013, (Araújo 2013), propôs o uso de uma nova métrica de similaridade chamada PICr em algoritmos de agrupamento. Essa nova métrica tem como base o PIC e foi utilizado nas funções custo de agrupamento (FCAs) com o objetivo de calcular o grau de interação entre as diferentes partes do conjunto de dados definidas pelas regiões auxiliares previamente encontradas. Nesse trabalho, três algoritmos de agrupamento baseados em teoria de informação também foram desenvolvidos para serem usados em conjunto com as FCAs. Os resultados foram bem satisfatórios tanto em contexto artificiais quanto em situações reais onde a complexidade dos dados é bem maior.

O estimador de entropia quadrática foi empregado na medida de divergências entre as densidades de probabilidade e separação às cegas em (Hild et al. 2006). Em (Principe et al. 2000) , expressões quadráticas com propriedades de informação mútua foram introduzidas com base em distâncias euclidianas e de Cauchy-Schwartz. Esse método apresentou simplicidade computacional e estabilidade estatística na otimização.

Em (Martins et al. 2014), o autor propôs uma nova medida chamada de QRNS a qual depende da estimativa da negentropia entre a Gaussiana que modela todos os clusters unidos e pela mistura de gaussianas construída pelos clusters individualmente. A QRNS é calculada em função da entropia de Renyi ao invés de Shannon.

Em (Banks et al. 2018) estudou-se o problema de detectar uma matriz de sinal estruturada e de baixa classificação corrompida com ruído gaussiano aditivo. Isso inclui clustering em um modelo de mistura Gaussiana, PCA esparsa e localização de submatriz.

Em (Lei et al. 2017) analisou-se o desempenho de oito índices populares de validade de clusters baseados em informações teóricas para partições encontradas por fuzzy c-means (FCM).

Em (Şeref et al. 2018) propõe-se uma nova estrutura computacional que integre a seleção de características baseadas em teoria da informação com o agrupamento k -median discreto.

De uma forma geral, pôde-se perceber que os trabalhos desenvolvidos apresentaram bons resultados e superaram as abordagens tradicionais em diversos contextos, o que demonstra robustez em dados com naturezas distintas. A partir disso, podemos concluir que o uso da teoria da informação vem trazendo grandes avanços no que diz respeito à recuperação da informação para guiar o processo de aprendizagem.

Devido aos grandes avanços em hardware e software para aquisição de dados ocorridos nos últimos anos, houve um aumento significativo nesse desse tipo de aplicação e consequentemente de pesquisas envolvendo o agrupamento de streams de dados. Com o advento desse tipo de aplicação, alguns métodos para “clusterização” de streams de dados foram propostos na literatura nos últimos anos. (Silva et al. 2013) fazem uma descrição do estado da arte do agrupamento de streams de dados, apresentando algoritmos, aplicações, softwares e repositórios de dados relacionados a esse tema. Dentre os algoritmos para agrupamento de streams de dados existentes, podemos destacar: STREAM (Guha et al. 2000), CluStream (Aggarwal 2003), DenStream (Cao et al. 2006) e AnyNovel (Abdallah et al. 2016). Esses algoritmos são geralmente baseados em algum algoritmo tradicional de “clusterização” de dados.

Em (Carnein et al. 2017) realizou-se um estudo de comparação entre 10 populares algoritmos de clustering de stream de dados. Utilizou-se vários conjuntos de dados reais e sintéticos e identificou-se as principais fraquezas e pontos fortes dos algoritmos estudados.

Em (Silva et al. 2017) desenvolveu-se um algoritmo de evolutivo baseado no k-means para agrupamento de stream de dados (FEAC-Stream) que permite estimar k automaticamente a partir dos dados de uma maneira online.

Ainda no contexto de clustering dinâmico, podemos citar trabalhos que se dedicaram a estudar critérios que otimizem as operações de fusão e separação entre os clusters. Entre eles, está (Lughofer 2012), onde foram propostos novos critérios para estas operações entre clusters dinâmicos. O autor sugeriu o uso do critério de homogeneidade e de toque entre clusters para a operação de fusão (merge). Para a divisão fez uso do critério de informação bayesiana. Os resultados do trabalho produziram partições mais confiáveis que as técnicas convencionais.

Em (Beringer & Hullermeier 2006) e (Beringer & Hullermeier 2007), foi apresentada uma técnica que aborda essas questões de fusão e separação integrando um conceito de divisão e fusão em bloco em um algoritmo *k*-means, onde as partições fundidas e divididas são comparadas com as partições baseadas em uma versão estendida do índice de validação de (Xie & Beni 1991); no entanto, para cada bloco de dados completo, apenas um cluster é mesclado ou dividido, ou seja, o número de clusters k aumenta ou diminui em 1.

Em (Song & Wang 2005), clusters na forma de modelos de mistura de gaussianas (GMMs) são treinados on-line por mecanismos de aprendizado incremental, integrando conceitos de fusão que (1) são aplicáveis apenas em chunkmode (novas gaussianas são geradas a partir de cada nova chegada de bloco de dados) e (2) requerem tempo de computação significativo, pois cada nova gaussiana é avaliada para a fusão com qualquer uma das gaussianas existentes. A divisão entre clusters não foi considerada em (Song & Wang 2005).

Os GMMs incrementais on-line em (Hall & Hicks 2005) Hall e Hicks (2005) aplicam ambos, um conceito de mesclagem e um conceito de divisão no modo incremental; no entanto, esse método é muito lento para aprendizado rápido on-line por causa de um agrupamento de gaussianas usando o limite de Chernoff. Uma aceleração é alcançada em (Declercq & Piater 2008), onde uma adequação para a fusão é avaliada com base na fidelidade de uma gaussiana usando o teste de Kolmogorov-Smirnoff. No entanto, esse método causa um aumento desnecessário na complexidade, porque cada nova amostra cria uma nova gaussiana, que geralmente requer uma re-mesclagem com gaussianas existentes que são mais significativas.

Todos esses trabalhos na área de clustering dinâmico apresentam bons resultados, porém são baseados em medidas de proximidade tradicionais, as quais normalmente crescem com o quadrado da distância, o que traz limitações ao método. Diante disso, o objetivo desse trabalho é incorporar as medidas da teoria da informação, já bastante estudadas no contexto de clustering simples, ao contexto do clustering dinâmico. A principal vantagem disso é o fato dessas medidas de informação estarem diretamente relacionadas à função de densidade de probabilidade dos dados, sem se fazer suposições como

gaussianidade ou isotropia dos clusters.

1.4 Organização do texto

O texto dessa tese está oganizado em 5 capítulos. Neste Capítulo 1 foi caracterizado o problema a ser abordado na Tese: Uso de descritores da teoria da informação no contexto de processos dinâmicos. Além disso, também foram traçados os objetivos e as contribuições da tese. No capítulo 2 são apresentados os conceitos principais da teoria da informação e da técnica de aprendizado por teoria da informação (ITL). No capítulo 3 são descritos os conceitos de clustering e seus os principais métodos. Além disso, também são apresentados os conceitos de clustering dinâmico e os principais algoritmos utilizados para esse fim. O capítulo 4, apresentamos a ideia central desse trabalho: o estudo da dinâmica da informação. Esse capítulo descreve a metodologia utilizada no trabalho, detalhando os casos de estudo investigados para se alcançar os diferentes objetivos dessa tese. Já o capítulo 5, descreve a metodologia e os resultados do trabalho no contexto dos clusters dinamicos. Finalmente, no Capítulo 6 são discutidas, com base nos resultados obtidos, algumas conclusões a respeito do trabalho, como também são propostas algumas ideias para trabalhos futuros.

Capítulo 2

Fundamentos da Teoria da Informação

Um problema comum enfrentado por muitos profissionais de processamento de dados é como encontrar a melhor maneira de extrair as informações contidas nos dados. Em nossa vida diária e em nossas profissões, somos bombardeados por uma enorme quantidade de dados, que na maioria das vezes não são do nosso interesse primário.

Os dados escondem, em termos de estrutura temporal ou em redundância espacial, pistas importantes para realizar o processamento de informações necessário para responder aos questionamentos levantados. (Principe 2010)

A teoria da informação é um ramo da matemática que estuda quantificação da informação. Essa teoria teve seus pilares estabelecidos por Claude Shannon em 1948 que formalizou conceitos com aplicações na teoria da comunicação e estatística. Segundo (Cover & Thomas 2006), a teoria da informação foi desenvolvida originalmente para compressão de dados, para transmissão e armazenamento. Em outras palavras, foi criada para ajudar a responder as questões teóricas de como codificar otimamente mensagens de acordo com suas estruturas estatísticas. Porém, foi planejada para aplicação ampla e têm sido usada em muitas outras áreas cujo fator de análise são os dados.

Resumidamente, a teoria da informação define medidas que tem o objetivo de quantificar a informação contida em uma dada variável aleatória ou até mesmo o montante de informações que uma variável aleatória possui sobre outra.

2.1 Aprendizado por Teoria da Informação (ITL)

Segundo (Principe 2010), o aprendizado por teoria da informação (ITL) é uma área de estudo que usa descritores da teoria da informação estimados a partir dos dados para substituir descritores estatísticos convencionais como variância e covariância. Recentemente, as técnicas de ITL vem sendo utilizadas com bastante sucesso em problemas de aprendizado não supervisionado. O grande sucesso do uso de ITL deve-se principalmente ao fato de ser uma técnica baseada em estatística de alta ordem, já que seus descritores são estimados diretamente a partir das amostras de dados. ITL propõe o uso de uma função de custo com eficiência computacional e sem sofrer a limitação de gaussianidade inerente às funções de custo baseadas em momentos de segunda ordem como o MSE. Isto é conseguido com o uso de descritores da teoria da informação como a entropia e medidas de dissimilaridades (divergência e informação mútua) combinados com estimadores de

função densidade de probabilidade (PDF) não paramétricos, que trazem robustez e generalidade para a função de custo e melhoram o desempenho em muitos cenários realistas.

2.1.1 Entropia

A entropia é o descritor mais conhecido da teoria da informação e o seu conceito foi primeiro utilizado na termodinâmica e posteriormente dado uma interpretação probabilística por Boltzmann em 1877 seguido por Plank em 1906. Shannon em 1928, foi um dos primeiros a aplicar o conceito de entropia a estudos em teoria da codificação e serviu de base para a atual teoria da informação.

Em 1948, Shannon definiu a Entropia de um conjunto X como sendo a soma das incertezas de todas as mensagens ponderadas pela probabilidade de cada uma delas. A entropia de Shannon para uma variável aleatória contínua X dada pela equação 2.1.

$$H(X) = -\int p(x) \log p(x) \, \mathrm{d}x \tag{2.1}$$

em que $p(x)$ é a pdf da V.A. X em questão.

A entropia H(X) é uma medida de quantidade média de informação transmitida por uma mensagem. Uma característica importante da medida é o fato de a combinação das incertezas serem ponderadas pelas suas probabilidades, que faz a essência do conceito de entropia. Segundo (Principe 2010), a relação entre a quantidade de informação de uma VA e a sua probabilidade de ocorrência é inversamente proporcional. Isso quer dizer que quanto mais raro for o evento, mais informação o mesmo conterá. Dessa forma, a entropia H(X) representa a soma das incertezas de todas as mensagens ponderadas pela probabilidade de cada uma delas.

2.1.2 Entropia de Renyi

A entropia de Shannon ocupa um papel central nos estudos da teoria da informação. No entanto, o conceito de informação é tão rico que talvez não haja uma definição única que seja capaz de quantificar informações devidamente. Além disso, do ponto de vista de engenharia, estimar entropia a partir dos dados não é uma questão trivial (Principe 2010).

Em 1950, Alfred Renyi introduziu uma família paramétrica de entropias como uma generalização matemática da entropia de Shannon conhecida como entropia de Renyi de ordem α, apresentada na equação 2.2.

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int p^\alpha(x) \, \mathrm{d}x \tag{2.2}$$

em que $\alpha >= 0$.

A entropia quadrática de Renyi $H_2(X)$ apresenta interesse particular devido à possibilidade de estimarmos a função densidade de probabilidade de X diretamente das amostras

por meio do uso de um método não paramétrico como as janelas de Parzen. A equação 2.3 define $H_2(X)$.

$$H_2(X) = -\log \int p^2(x)\,\mathrm{d}x \tag{2.3}$$

O argumento do logaritmo na equação 2.3 é conhecido como Potencial de Informação (IP), o qual é uma medida de informação também muito utilizada atualmente.

$$V_2(X) = \int_{-\infty}^{\infty} p^2(x)\,\mathrm{d}x \tag{2.4}$$

A estimativa da pdf de X, segundo o método de Parzen, utilizando uma função kernel arbitrária $K_\sigma(.)$ é dada por:

$$\hat{p}_X(x) = \frac{1}{N\sigma}\sum_{i=1}^{N} K\left(\frac{x-x_i}{\sigma}\right) \tag{2.5}$$

em que σ é o *kernel size* e $\{x_1, x_2, x_3, ..., x_N\}$ são as N amostras da variável aleatória X.

A partir das equações 2.3 e 2.5 e assumindo um kernel gaussiano $G_\sigma(.)$ com desvio padrão σ, obtemos uma nova equação para a estimação da entropia quadrática de Renyi.

$$\hat{H}_2(X) = -\log \int \left(\frac{1}{N}\sum_{i=1}^{N} G_\sigma(x-x_i)\right)^2 \mathrm{d}x \tag{2.6}$$

Como a integral em 2.6 não é de simples resolução, são realizadas algumas manipulações algébricas a fim de reescrever a equação:

$$\hat{H}_2(X) = -\log\frac{1}{N^2}\int \left(\sum_{i=1}^{N}\sum_{j=1}^{N} G_\sigma(x-x_j)\cdot G_\sigma(x-x_i)\right)\mathrm{d}x$$

$$\hat{H}_2(X) = -\log\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\int G_\sigma(x-x_j)\cdot G_\sigma(x-x_i)\,\mathrm{d}x$$

$$\hat{H}_2(X) = -\log\left(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_j-x_i)\right) \tag{2.7}$$

onde

$$G_\sigma(\mu) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} exp - \frac{||\mu||^2}{2\sigma^2}$$

Logo, temos em 2.7, uma expressão analítica para estimar a entropia de uma VA usando apenas suas amostras. O potencial de informação é apresentado na equação 2.8.

$$\hat{H}_2(X) = -\log(\hat{V}_2(X))$$

$$\hat{V}_{2,\sigma}(X) = \left(\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_j - x_i) \right) \tag{2.8}$$

Segundo (Principe et al. 2000), a equação 2.7 representa um dos principais resultados do aprendizado por teoria da informação, pois mostra que o IP, pode ser estimado diretamente a partir das amostras com um cálculo exato da integral sobre a variável aleatória para kernels gaussianos.

Esse capítulo apresentou os conceitos das medidas da ITL mais conhecidas e que são necessárias ao desenvolvimento desse trabalho. O capítulo seguinte apresentará os conceitos da técnica de clustering, em particular o clustering dinâmico, o qual foi o foco da terceira fase desse trabalho.

Capítulo 3

Fundamentos de Clustering

Uma das mais importantes das inúmeras atividades de análise de dados é classificar ou agrupar dados em um conjunto de categorias ou clusters. Algoritmos de clusterização particionam objetos de dados (padrões, entidades, instâncias, observâncias, unidades) em um determinado número de clusters (grupos, subconjuntos ou categorias). No entanto, não há uma definição universal e precisa do termo cluster.

Claramente, um cluster em todas as definições é descrito em termos de homogeneidade interna e separação externa (Gordon 1999) (Jain & Dubes 1988), ou seja, objetos de dados dentro de um cluster devem ter alta semelhança uns com os outros, mas devem ser muito diferentes de objetos em outros clusters. As diferenças são avaliadas com base nos valores dos atributos que descrevem esses objetos. Tanto a semelhança como a dissimilaridade devem ser definidos de forma clara e significativa. Muitas vezes, medidas de distância são usadas.

O processo de clustering é útil em várias situações exploratórias da análise de padrões, agrupamento, tomada de decisão e aprendizado de máquina, incluindo mineração de dados, recuperação de documentos, segmentação de imagens e classificação de padrões. No entanto, em muitos desses problemas, há pouca informação prévia (por exemplo, modelos estatísticos) disponível sobre os dados, e o tomador de decisão deve fazer o mínimo possível de suposições sobre os dados. É sob essas restrições que a metodologia de agrupamento é uma forma de aprendizado por observação, ao invés de aprendizado por exemplos.

Segundo (Xu & Wunsch II 2009), a atividade típica de agrupamento de padrões envolve alguns passos que podem ser observados na figura 3.1 e são descritos a seguir:

- Seleção ou extração de características: A seleção de características escolhe informações distintas de um conjunto de candidatos, enquanto a extração de características utiliza algumas transformações para gerar recursos úteis e inovadores a partir dos originais. Claramente, a extração de características é potencialmente capaz de produzir recursos que poderiam ser mais úteis na descoberta da estrutura de dados. (Jain & Dubes 1988)

- Seleção de algoritmos de clustering: Essa etapa geralmente consiste em determinar uma medida de proximidade apropriada e construir uma função de critério. Uma vez que uma medida de proximidade foi determinada, o clustering pode ser considerado como um problema de otimização com uma função específica de critério.

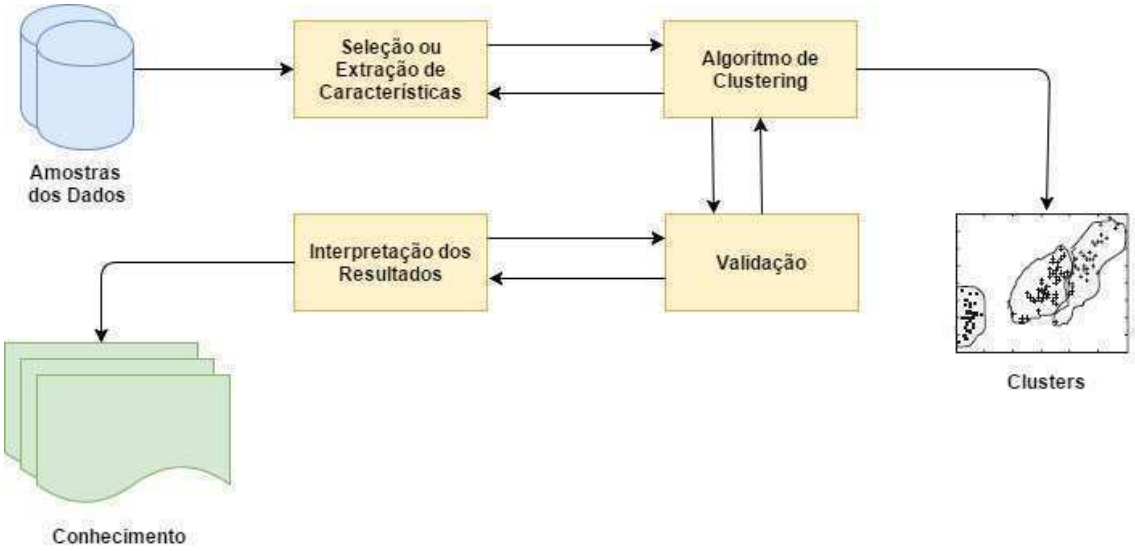


Figura 3.1: Etapas na Análise de Cluster. Fonte: Clustering. Rui Xu e Donald Wunsch II, Wiley, 2009.

Novamente, os clusters obtidos dependem da seleção da função de critério. A subjetividade da análise de cluster é, portanto, inevitável. Uma grande quantidade de algoritmos de clustering foi desenvolvida para resolver diferentes problemas de uma ampla variedade de campos. No entanto, não existe um algoritmo de agrupamento universal para resolver todos os problemas.

- Validação de Cluster: Dado um conjunto de dados, cada algoritmo de agrupamento sempre pode produzir uma partição, quer exista ou não uma estrutura particular nos dados. Além disso, diferentes abordagens de cluster geralmente levam a diferentes clusters de dados e, para um mesmo algoritmo, a seleção de um parâmetro ou a ordem de apresentação dos padrões de entrada pode afetar os resultados finais. Portanto, os critérios de avaliação eficazes são extremamente importantes para fornecer aos usuários um grau de confiança para os resultados do agrupamento.

- Interpretação dos resultados: O objetivo final do agrupamento é fornecer aos usuários informações significativas a partir dos dados originais, para que eles possam desenvolver um entendimento claro dos dados e, portanto, resolver efetivamente os problemas encontrados.

3.1 Categorização dos Principais Métodos de Clustering

Muitos algoritmos de clustering existem na literatura. É difícil fornecer uma categorização nítida dos métodos de clustering porque essas categorias podem se sobrepor, de modo que um método pode ter recursos de várias categorias.

Em geral, os principais métodos de agrupamento podem ser classificados nas seguintes categorias.

- Métodos de particionamento: Dado um banco de dados com n objetos ou tuplas de dados, um método de particionamento constrói k partições dos dados, onde cada partição representa um cluster e $k <= n$. Ou seja, classifica os dados em k grupos, que juntos satisfazem os seguintes requisitos: (1) cada grupo deve conter pelo menos um objeto, e (2) cada objeto deve pertencer a exatamente um grupo. Para alcançar a otimização global em clustering baseado em particionamento, seria necessário enumeração exaustiva de todas as partições possíveis. Em vez disso, a maioria das aplicações adotam uma das heurísticas populares, como (1) o algoritmo k-means, onde cada cluster é representado pelo valor médio dos objetos no cluster e (2) o algoritmo k-medoids, onde cada cluster é representado por um dos objetos localizado perto do centro do cluster. Esses métodos de agrupamento heurísticos funcionam bem para encontrar clusters de formato esférico em bancos de dados de pequeno e médio porte. Para encontrar clusters com formas complexas e para agrupar conjuntos de dados muito grandes, os métodos baseados em particionamento precisam ser estendidos.

- Métodos hierárquicos: Um método hierárquico cria uma decomposição hierárquica do conjunto dado de objetos de dados. Um método hierárquico pode ser classificado como sendo aglomerativa ou divisiva, com base na forma como a decomposição hierárquica é formada. A abordagem aglomerativa, também chamada de abordagem bottom-up, começa com cada objeto formando um grupo separado. Ele mescla sucessivamente os objetos ou grupos que estão próximos um ao outro, até que todos os grupos sejam fundidos em um (o nível mais alto do hierarquia) ou até que uma condição de rescisão seja mantida. A abordagem divisiva, também chamada de abordagem de top-down começa com todos os objetos no mesmo cluster. Em cada iteração, um cluster é dividido em clusters menores, até que, eventualmente, cada objeto esteja em um cluster, ou até que uma condição de finalização seja válida.

- Métodos baseados em densidade: a maioria dos métodos de particionamento agrupam objetos com base na distância entre os objetos. Tais métodos podem encontrar apenas clusters esféricos e ter dificuldade em descobrir clusters de formas arbitrárias. Outros métodos de clusterização foram desenvolvidos com base na noção de densidade. Sua ideia geral é continuar crescendo o cluster dado, desde que a densidade (número de objetos ou pontos de dados) na "vizinhança" exceda algum limite; isto é, para cada ponto de dados dentro de um determinado cluster, a vizinhança dentro de um dado raio deve conter pelo menos um número mínimo de pontos. Tal método pode ser usado para filtrar o ruído (outliers) e descubrir agrupamentos de forma arbitrária. Um exemplo de algoritmo que utiliza esse princípio é o DBSCAN (Ester et al. 1996).

- Métodos baseados em grades: esses métodos dividem o espaço de dados em uma grade, onde cada célula dessa grade pode ainda ser dividida em outra grade e assim sucessivamente, criando um espaço de dados com diferentes resoluções. As amostras são, então, agrupadas de acordo com as células criadas. Essas células podem ainda ser unidas umas com as outras. Ao final dos processo, cada célula represen-

tará um cluster. Dois exemplos de algoritmos que utilizam essa metodologia são o CLIQUE (Agrawal et al. 1998) e o MAFIA (Goil et al. 1999).

Alguns algoritmos de clustering integram as ideias de vários métodos de clustering de modo que, às vezes, é difícil classificar um determinado algoritmo como pertencendo exclusivamente a apenas uma categoria de método de agrupamento. Além disso, algumas aplicações podem ter critérios de agrupamento que requerem a integração de várias técnicas de agrupamento.

3.2 Clusters Dinâmicos

O problema de clustering de stream de dados trata de agrupar uma sequência de dados potencialmente infinita, não estacionária (a distribuição de probabilidade dos dados é desconhecida e pode mudar ao longo do tempo) e que chega continuamente (o que requer uma única passagem através dos dados). Neste caso, o acesso aleatório aos dados não é viável e armazenar todos os dados que chegam é impraticável.

Nas aplicações de hoje em dia, os streams de dados são onipresentes. De fato, exemplos de aplicações relevantes para streams de dados estão se tornando mais numerosos e mais importantes, incluindo detecção de intrusão em redes, fluxos de transações, registros de telefone e monitoramento do tempo. Há uma pesquisa ativa sobre como armazenar, consultar, analisar, extrair e prever informações relevantes de stream de dados.

Diante disso, há um aumento da demanda por modelos que podem automaticamente mudar a dinâmica dos processos em sistemas com alta complexidade e com características variáveis no tempo. A fim de abordar essa dinâmica, a metodologia de "evolving intelligent systems" (EIS) (Angelov et al. 2010) foi desenvolvida ao longo da última década e desempenham um papel importante, pois empregam abordagens e mecanismos que atualizam os parâmetros do modelo incrementalmente e desenvolvem novos componentes de modelo conforme necessário com base nos stream de dados registrados on line e que refletem as alterações do processo ao longo do tempo.

Evolving cluster models (ECM) é um importante sub-área dos evolving intelligent system e são projetados para explorar uma enorme quantidade de streams de dados on line (Bifet et al. 2010) ou bancos de dados muito grandes (VLDBs) por meio do agrupamento e da compressão de dados de uma forma rápida e incremental dentro de um contexto de aprendizado não supervisionado. Os ECMs também são referenciados como métodos de "incremental clustering" ou "evolving clustering" (Bouchachia 2011), porque eles podem processar etapas de dados, atualizar e evoluir partições de cluster em etapas de aprendizado incremental.

Abordagens tradicionais para clustering de dados não são adequadas para aplicações baseadas em streams de dados. Nessas aplicações, o conjunto de dados não está completamente disponível desde o início do processamento e os dados são obtidos ao longo do tempo, obedecendo uma determinada taxa de amostragem. Dessa forma, o conjunto de dados é tratado como sendo uma grande sequência de observações, onde cada amostra é apresentada ao algoritmo de agrupamento apenas após a sua aquisição. Já os métodos tradicionais de clustering trabalham geralmente de forma offline, ou seja, necessitam que

o conjunto de dados completo esteja disponível no inicio do processamento, o que não é possível quando estamos analisando streams de dados. (Bezerra 2017)

Ao aplicar técnicas de mineração de dados e, especificamente, algoritmos de cluster, a streams de dados, restrições no tempo de execução e memória devem ser consideradas com cuidado. Para lidar com restrições de tempo e memória, muitos dos algoritmos de clustering de fluxo de dados existentes modificaram o método tradicional não-streaming para usar a estrutura de duas fases para lidar com dados de streaming, por exemplo, DenStream é uma extensão do algoritmo DBSCAN, StreamKM ++ de k-means ++, StrAP de AP, entre outros.

Além disso, um algoritmo de clustering de streams de dados precisa se adaptar às mudanças que ocorrem na dinâmica dos dados analisados ao longo do seu processo de análise, já que esses dados são muitas vezes não estacionários. Também é de se esperar que o algoritmo possa lidar com clusters que não tenham um formato previamente definido.

Um problema específico em ECMs surge sempre que duas ou mais nuvens de dados locais (cada uma modelada por um cluster) estão se movendo juntas ou são separadas dentro de um cluster. Nesses casos, os clusters devem ser mesclados (fusão, merge) ou separados (divisão, split) dinamicamente para manter a alta qualidade nas partições de cluster que seguem a distribuição natural das nuvens de dados.

3.2.1 Algoritmos de Clustering de Streams de Dados

Devido aos grandes avanços em hardware e software para aquisição de dados ocorridos nos últimos anos, houve um aumento significativo nesse desse tipo de aplicação e consequentemente de pesquisas envolvendo o agrupamento de streams de dados (Silva et al. 2013). Com o advento desse tipo de aplicação, alguns métodos para clustering de streams de dados foram propostos na literatura nos últimos anos. (Silva et al. 2013) fazem uma descrição do estado da arte do agrupamento de streams de dados, apresentando algoritmos, aplicações, softwares e repositórios de dados relacionados a esse tema. Dentre os algoritmos para agrupamento de streams de dados existentes, podemos destacar: STREAM (Guha et al. 2000), CluStream (Aggarwal 2003), DenStream (Cao et al. 2006) e AnyNovel (Abdallah et al. 2016). Esses algoritmos são geralmente baseados em algum algoritmo tradicional de clustering de dados (Ghesmoune et al. 2016).

A maioria dos algoritmos existentes (por exemplo, CluStream , DenStream , StreamKM ++ ou ClusTree) divide o processo de agrupamento em duas fases: (a) Online, os dados são sumarizados; (b) Off-line, os clusters finais serão gerados. Figura 3.2 apresenta um fluxograma com alguns algoritmos de clustering de streams de dados mais utilizados.

STREAM: Um dos primeiros algoritmos desenvolvidos para realizar o agrupamento de streams de dados. O STREAM realiza o agrupamento particionando o stream de dados em sequências menores e aplica um algoritmo de particionamento semelhante ao k-means nessas sequências. Assim, esse algoritmo necessita que sejam armazenadas todas as amostras referentes a sequência para realizar o agrupamento, além de necessitar também do número de clusters antecipadamente.

E-Stream : classifica a evolução dos dados em cinco categorias: aparecimento, desa-

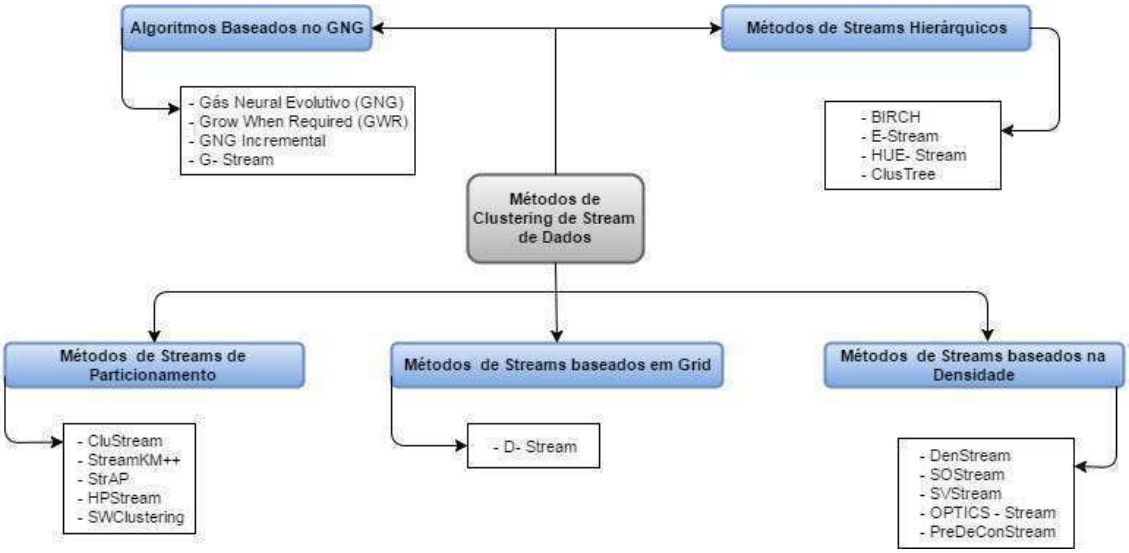


Figura 3.2: Taxonomia dos Algoritmos de Clustering de Stream de Dados. Fonte: State-of-the-art on clustering data streams. Ghesmoune et all, Big Data Analytics, 2016.

parecimento, auto evolução, fusão e divisão. Este algoritmo é um método de clustering de stream de dados baseado em evolução, ou seja, um método de clustering de streams de dados que suporta o monitoramento e a detecção de alterações nas estruturas dos cluster. Um histograma dos valores de dados do cluster é utilizado para identificar divisões de cluster. Se um vale estatisticamente significativo é encontrado entre dois picos em qualquer um dos histogramas, o cluster é dividido.

CluStream : divide o processo de clustering em um componente on-line que periodicamente armazena um resumo de estatísticas detalhadas e um componente off-line que usa apenas este resumo estatístico. A fase on-line armazena q micro-clusters na memória (secundária), onde q é um parâmetro de entrada. Para cada nova amostra é utilizado um algoritmo de particionamento semelhante ao k-means, para determinar a qual micro-cluster ela pertence. O CluStream mantém sempre armazenados uma quantidade fixa de micro-clusters, permitindo a criação e fusão de micro-clusters. Por fim, o CluStream realiza um outro agrupamento de forma offline, onde os micro-clusters são, então, agrupados em grupos definitivos utilizando o k-means.

StreamKM ++: é um algoritmo de duas fases (online-offline) que mantém um pequeno esboço dos dados de entrada usando a técnica de mesclar e reduzir.

DenStream: é um algoritmo que realiza o agrupamento das amostras baseado na sua densidade. Ele também trabalha com o conceito de micro-clusters, onde cada um desses micro-clusters possui um peso na geração dos clusters propriamente ditos. Um micro-cluster que não possui amostras recentes tem seu peso reduzido nesse processo, enquanto que os que possuem amostras recentes têm seus pesos aumentado. Para a obtenção dos clusters, o DenStream aplica aos micro-clusters um processo semelhante ao DBSCAN. O formato dos clusters obtidos é arbitrário, ou seja, não possuem uma forma definida. Uma desvantagem do DenStream, assim como do CluStream, é o número de parâmetros que precisam ser pré-definidos para o seu bom funcionamento.

SOStream: mescla, atualiza e adapta dinamicamente o valor de limite para cada cluster de maneira on-line. Os clusters são mesclados se eles se sobrepuserem a uma distância menor que o limite de mesclagem, ou seja, as esferas no espaço d-dimensional definido pelo raio de cada cluster se sobrepõem. Assim, o valor limite é um fator determinante fator para o número de clusters. No entanto, nenhum recurso de divisão é proposto no algoritmo.

A figura 3.3 apresenta uma tabela com um resumo das principais características dos algoritmos de clusterização de stream de dados.

Algorithms	Based on	Topology	WL	Phases	Remove	Merge	Split	Fade
SVStream	SVC, SVDD	✗	✗	online	✓	✓	✓	✓
StreamKM++	*k*-means++	✗	✗	2 phases	✓	✓	✓	✓
StrAP	AP	✗	✗	2 phases	✓	✗	✗	✓
SOStream	DBSCAN, SOM	✗	✗	online	✓	✓	✗	✓
OPTICS-Stream	OPTICS	✗	✗	2 phases	✓	✓	✗	✓
IGNG	NG	✓	✗	online	✗	✗	✗	✗
HCluStream	*k*-prototypes	✗	✗	2 phases	✓	✓	✓	✓
GWR	NG	✓	✗	online	✗	✗	✗	✗
G-Stream	NG	✓	✓	online	✓	✗	✗	✓
E-Stream	*k*-means	✗	✗	2 phases	✓	✓	✓	✓
D-Stream	-	✗	✗	2 phases	✓	✓	✓	✓
DenStream	DBSCAN	✗	✗	2 phases	✓	offline	✗	✓
ClusTree	*k*-means or DBSCAN	✗	✗	2 phases	✓	offline	✓	✓
CluStream	*k*-means	✗	✗	2 phases	✓	offline	✗	✗
AING	NG	✓	✗	online	✗	✓	✗	✗

Figura 3.3: Características dos Algoritmos de Clustering de Stream de Dados. Fonte: State-of-the-art on clustering data streams. Ghesmoune et all, Big Data Analytics, 2016.

Note que a maioria dos algoritmos citados realizam o clustering em duas fases. O clustering de duas fases separa o processo de agrupamento em componentes on-line e off-line. A fase on-line constroi micro-clusters e captura informações resumidas a partir do stream de dados, e a fase offline gera os clusters finais a partir dos micro-clusters. A fase off-line geralmente é baseada em algum algoritmo tradicional de clustering.

Independente da tecnica utilizada, muitos desses algoritmos permitem as operações de junção e separação entre os clusters ao longo do tempo. Como dito antes, essas operações são muito valorizadas em um algoritmo para streams de dados. Entretanto, cada um desses algoritmos realiza essas operações de maneira diferente, seguindo critérios próprios.

Alguns algoritmos realizam as operações dejunção e separação baseadas nas mudanças das distribuições dos dados. Outros algoritmos utilizam critérios geométricos para estas operações, baseados em alguma medida de similaridade, indicando uma sobreposição de nuvens de dados ou um distanciamento entre clusters. Além disso, a homogeneidade dos clusters também é levada em consideração para os casos de junção. Normalmente, um cluster é dividido quando sua variância está acima de um limiar pré-estabelecido e dois clusters são mesclados quando a distância entre seus centróides está abaixo de outro

limite pré-especificado. Diante disso, algumas das principais medidas de similaridades que servem de base nessas operações serão apresentadas na próxima seção.

3.3 Medidas de Proximidade

Segundo (Xu & Wunsch II 2009), clusters são considerados como grupos contendo objetos de dados que são semelhantes uns aos outros, enquanto os objetos de dados em clusters diferentes não são. Assim, é natural perguntar que tipo de padrões devemos usar para determinar a proximidade, ou como medir a distância (dissimilaridade) ou similaridade entre um par de objetos, um objeto e um cluster ou um par de clusters. Medidas de similaridade (ou dissimilaridade) servem para comparar dois ou mais padrões em um processo de classificação ou medir a pertinência de um dado padrão a uma classe qualquer.

Proximidade é a generalização de ambos dissimilaridade e similaridade. Uma dissimilaridade ou função de distância em um conjunto de dados X é definido para satisfazer as seguinte condições:

1. Simetria
   $D(x_i, x_j) = D(x_j, x_i)$

2. Positividade
   $D(x_i, x_j) >= 0 \qquad \forall\, x_i \, e \, x_j$

   Se as condições

3. Desigualdade triangular
   $D(x_i, x_j) <= D(x_i, x_k) + D(x_k, x_j) \qquad \forall\, x_i,\ x_j e\, x_k$
   e

4. Reflexividade
   $D(x_i, x_j) = 0 \qquad sse\ \ x_i = x_j$

também se mantiverem, a função é chamada de métrica. Se apenas a desigualdade triangular não for satisfeita, a função é chamada de semimétrica.

Da mesma forma, uma função de similaridade é definida para satisfazer as condições abaixo:

1. Simetria
   $S(x_i, x_j) = S(x_j, x_i)$

2. Positividade
   $0 <= S(x_i, x_j) <= 1 \qquad \forall\, x_i \, e \, x_j$

   se também satisfizer as seguintes condições adicionais

3. $\forall\, x_i,\ x_j e\ x_k$
   $S(x_i,x_j)S(x_j,x_k) <= [S(x_i,x_j)+S(x_j,x_k)]S(x_i,x_k)$
   e

4. $S(x_i,x_j) = 1 \qquad sse\ \ x_i = x_j$

É chamada de métrica de similaridade.

3.3.1 Medidas Tradicionais de Proximidade

Distância Euclidiana

Talvez a medida de distância mais usada seja a distância euclidiana, também conhecida como norma L2, representada como

$$D(x_i,x_j) = (\sum_{l=1}^{d} |x_{il} - x_{jl}|^{\frac{1}{2}})^2 \tag{3.1}$$

onde x_i e x_j são objetos de dados d-dimensionais. Não é difícil ver que a distância euclidiana satisfaz todas as condições na seção 3.3, sendo portanto uma métrica.

Sabe-se que a distância euclidiana tende a formar clusters hiperesféricos. Além disso, clusters formados com a distância euclidiana são invariantes a translações e rotações no espaço de características. No entanto, se os atributos forem medidos com unidades bastante diferentes, características com grandes valores e variâncias tenderão a dominar outras características. Transformações lineares ou outras transformações também podem causar distorção nas relações de distância. Uma maneira possível de lidar com esse problema é normalizar os dados para que cada recurso contribua igualmente para a distância.

Distância Minkowski

A distância euclidiana pode ser generalizada como um caso especial de uma família de métricas, chamadas distância Minkowski ou norma *Lp*, definidas como,

$$D(x_i,x_j) = (\sum_{l=1}^{d} |x_{il} - x_{jl}|^{\frac{1}{p}})^p \tag{3.2}$$

Note que quando $p = 2$, a distância se torna a distância Euclidiana.

Distância City-block e Distância Sup

Fazendo $p = 1$ e $p = \infty$ na equação 3.2, obtem-se dois casos especiais da distância de Minkowski: a city-block, também conhecida como distância Manhattan ou norma *L1*, e a distância Sup ou norma L_∞.

$$D(x_i,x_j) = (\sum_{l=1}^{d} |x_{il} - x_{jl}|) \tag{3.3}$$

$$D(x_i, x_j) = max_{1<=l<+d}|x_{il} - x_{jl}| \tag{3.4}$$

Distância de Mahalanobis

A distância quadrática de Mahalanobis também é uma métrica e é definida como

$$D(x_i, x_j) = (x_i - x_j)^T S^{-1}(x_i - x_j) \tag{3.5}$$

onde S é a matriz de covariância dentro da classe definida.

A distância de Mahalanobis tende a formar clusters hiperelipsoidais, são invariantes a qualquer transformação linear não singular. No entanto, o cálculo da inversa de S pode causar alguma carga computacional para dados em grande escala. Quando os atributos não estão correlacionados, o que leva S a uma matriz de identidade, a distância quadrática de Mahalanobis é equivalente à distância euclidiana ao quadrado (Jain & Dubes 1988).

3.3.2 Medidas de Proximidade Não-Euclidianas

Em geral há sempre uma estatística envolvida na distribuição dos padrões em um conjunto de dados e cada classe (cluster) possui sua distribuição. A utilização de uma métrica que incorpore esta estatística apresenta uma vantagem em relação a medida de distância euclidiana comum. Como mostra o capítulo 2, as medidas da teoria da informação estão intimamente ligadas à estatística de alta ordem, por esse motivo, apresentaremos as principais medidas de proximidade baseadas na teoria da informação.

Divergência de Kullback-Leibler

Em 1936, Mahalanobis introduziu pela primeira vez o conceito de "distância"entre duas distribuições de probabilidade, e desde então muitos resultados importantes foram estabelecidos. Vamos considerar duas diferentes distribuições de probabilidades p(x) e q(x), e definir a divergência de Kullback-Leibler (KL) como

$$D_{KL}(p||q) = \sum p(x) \log \frac{p(x)}{q(x)} \tag{3.6}$$

A divergência de KL está efetivamente medindo a dissimilaridade entre dois modelos de distribuição p e q para uma dada VA X. No entanto, não podemos chama-lá de distância porque obedece apenas a um dos postulados de distância. Por essa razão, é chamada de divergência. Como, em tese, cada cluster é modelado por uma distribuição de probabilidades distinta, a divergência KL pode ser utilizada como medida de distância entre os clusters. Para isso, necessita-se conhecer as distribuições de probabilidade de cada cluster, as quais podem ser obtidas por meio de métodos de estimação como as janelas de Parzen.

Divergência utilizando entropia de Renyi

A divergência entre duas distribuições pode ser vista como uma medida baseada na entropia de cada distribuição. Diante disso, pode-se utilizar a entropia de Renyi para calcular a divergência entre duas distribuições. Em (Martins 2005), fez-se uso da entropia de Renyi para medir a divergência entre dois clusters conforme equação abaixo:

$$D_r(p(x)||q(x)) = H_r(X,Y) - H_r(X_g) \tag{3.7}$$

Na expressão, X,Y é o conjunto formado pelos pontos dos cluster X e Y. X_g é um modelo de distribuição gaussiana estimada utilizando o conjunto inteiro X,Y. Nesse caso, compara-se a "quantidade de variância"presente no conjunto todo, com a variância de uma gaussiana que modela o mesmo. Quando os clusters se distanciam, a variância da gaussiana equivalente aumentará, aumentando assim a divergência.

Potencial de Informação Cruzado - PIC

Da mesma forma que o potencial de informação foi definido em termos da entropia, (Gokcay 2000) nomeou a parte principal da estimativa de entropia cruzada de Renyi como sendo o potencial de informação cruzado. A entropia cruzada de Renyi é definida como

$$H_2(X;Y) = -log(\int f(x)g(x)dx) \tag{3.8}$$

e o PIC é definido como

$$V(p,q) = \int f(x)g(x)dx \tag{3.9}$$

em que $p(x)$ e $q(x)$ são duas funções densidade de probabilidade. Quando as distribuições são iguais, o PIC tem o mesmo valor que o IP.

O potencial de informação cruzado tem sido bastante utilizado em trabalhos que envolvem a tarefa de agrupamento. Segundo (Araújo 2013), se pensarmos nos grupos como distribuições de probabilidade, podemos usar o PIC para medir a interação entre as diferentes distribuições, o que o torna uma poderosa medida de similaridade no processo de clustering em conjuntos de dados complexos.

Negentropia

A negentropia entre duas distribuições consiste em medir a divergência entre a distribuição de probabilidade de dois clusters juntos, com o modelo gaussiano equivalente. Essa medida é definida como:

$$J = H_g(p_x(x)) - H(p_x(x)) \tag{3.10}$$

em que $H(p_X(x))$ é a entropia da pdf da VA x e $H_g(p_x(x))$ é a entropia da VA gaussiana com mesma variância de $p_x(x)$.

A negentropia é uma medida não-negativa e se torna zero quando a VA X tem distribuição gaussiana. Em outras palavras, a negentropia mede o quanto a distribuição de uma

VA não se parece com uma gaussiana.

Separação por Negentropia Quadrática de Renyi - QRNS

A QRNS baseia-se na estimativa da negentropia entre os clusters modelados como uma gaussiana e a mistura de gaussianas construída pelos clusters individualmente. Essa medida foi proposta em (Martins et al. 2014) como uma forma analítica para estimar a divergência entre clusters usando a entropia quadrática de Rényi ao invés da entropia de Shannon. Em (Martins et al. 2014), a QRNS foi utilizada como criterio na decisão de unir ou não pequenos clusters gaussianos após quantização vetorial no conjunto de entrada com o intuito de formar clusters mais complexos. A definição de negentropia de uma mistura é dada por:

$$J = H(g_x(x)) - H(p_x(x)) \tag{3.11}$$

em que $H(g_x(x))$ é a entropia da aproximação gaussiana do modelo da mistura e $H(p_x(x))$ é a entropia da mistura. Formalmente temos que:

$$p_x(x) = P_1 N(x, \mu_1, \Sigma_1) = P_2 N(x, \mu_2, \Sigma_2) \tag{3.12}$$

e

$$g_x(x) = N(x, \mu_q, \Sigma_q) \tag{3.13}$$

em que $N(x, \mu_i, \Sigma_i)$ é uma gaussiana com seus parâmetros e P1 e P2 são as probabilidades a priori de cada gaussiana na mistura. O índice q é usado para indicar a mistura de parâmetros e é dado por:

$$\mu_q = P1\mu_1 + P2\mu_2 \tag{3.14}$$

$$\Sigma_q = P_1\Sigma_1 + P_2\Sigma_2 + P_1P_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)^t \tag{3.15}$$

Diante disso, a QRNS é expressa por:

$$QRNS = \log\left(\frac{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(P_1N(x,\mu_1,\Sigma_1) + P_2N(x,\mu_2,\Sigma_2))^2 dx}{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}N(x,\mu_q,\Sigma_q)^2 dx}\right) \tag{3.16}$$

Esse capítulo apresentou o conceito de clustering e de clustering dinâmico, bem como as principais técnicas utilizadas nessas áreas. Foram apresentados também conceitos e as principais medidas de proximidade utilizadas no processo de clustering, Euclidianas e não-Euclidianas.

Capítulo 4

Dinâmica da Informação

O termo processos estocásticos é usado para descrever a evolução no tempo de um fenômeno aleatório de acordo com leis probabilísticas (Haykin 2013). De certa forma, representam uma extensão do conceito de variável aleatória para processos dinâmicos.

Segundo (Campos et al. 2016), um processo estocástico é uma família de variáveis aleatórias que evoluem com o tempo. Isto é, um processo estocástico X é um conjunto $\{X_t : t \in T\}$, onde cada X_t é uma variável aleatória e T é um conjunto totalmente ordenado que representa o tempo. A figura 4.1 apresenta um esquema geral de um processo estocástico $X(t)$, onde cada $X(t_i)$ representa uma variável aleatória e cada ξ_j representa o resultado de um experimento aleatório.

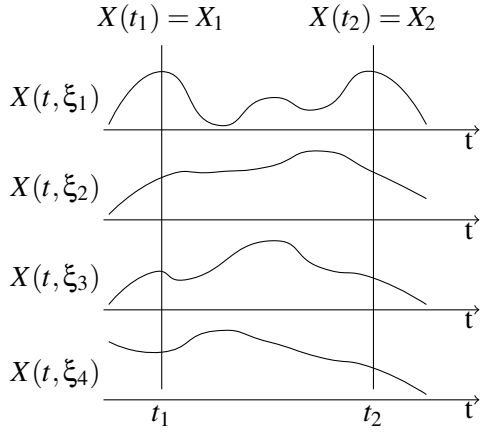


Figura 4.1: Esquema Geral de um Processo Estocástico.
(Papoulis & Pillai 2002)

Com a evolução do tempo, as propriedades estatísticas do processo, tais como média, variância e função densidade de probabilidade (fdp) podem sofrer alterações. Assim como as medidas estatísticas convencionais, as medidas de informação do processo, tais como entropia e potencial de informação, também são alteradas com o tempo. Isso acontece porque esses descritores medem a informação contida em variáveis aleatórias e estas variáveis estão evoluindo no tempo.

Um dos objetivos desse trabalho é investigar a forma com que essas medidas de informação mudam com o decorrer do tempo, ou seja, descobrir a dinâmica que rege o

comportamento da informação. Entende-se por comportamento dinâmico da informação a variação espaço temporal da informação, quantificada por medidas da teoria da informação. Inicialmente, o estudo dedica-se a comprovar a existência de dinâmica na informação ao longo do tempo, sendo essa a primeira contribuição desse trabalho. Para isso faz-se uso de vídeos como exemplo de ambiente dinâmico e faz-se uma análise do potencial de informação de cada frame à medida que o vídeo evolui no tempo. Para a identificação da dinâmica presente na informação, faz-se uso de uma rede neural autorregressiva (NAR).

Diante do sucesso obtido no estudo da dinâmica da informação, uma segunda contribuição desse trabalho consiste numa representação espaço-estados para processos dinâmicos baseada na sua informação. Chamamos essa representação de Modelo Estados de Informação, o qual consiste em um modelo espaço-estados onde as variáveis de estado são medidas de informação do sistema. Para a validação do modelo, é implementada uma aplicação novamente utilizando vídeos como exemplo de ambientes dinâmicos. O objetivo da aplicação é avaliar a qualidade dos vídeos por meio de sua informação.

4.1 Presença de Dinâmica na Informação

Os vídeos são exemplos de ambientes estocásticos e por esse motivo foram escolhidos para serem objeto de estudo nessa primeira fase do trabalho onde o objetivo principal é a comprovação da existência de dinâmica na informação. Para isso, considere o seguinte cenário: Um vídeo com n frames, onde cada frame do vídeo possui w pixels com informações das cores presentes na imagem que o compõem. Esses pixels podem ser considerados amostras de uma dada variável aleatória X que evolui no tempo. Desse modo, esse cenário nos fornece uma fonte de dados da variável aleatória X a cada instante de tempo, o que nos permite estimar diferentes medidas de informação para cada instante, entre elas o IP. A figura 4.2 apresenta um esquema para esse cenário, onde frames de vídeos representam as variáveis aleatórias em um processo estocástico.

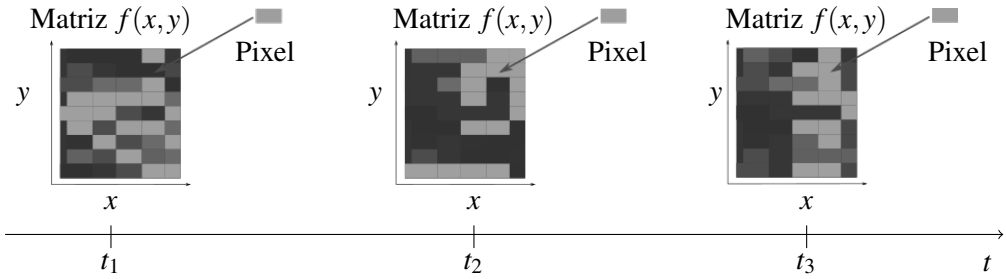


Figura 4.2: Vídeo como exemplo de processo estocástico

Nesse contexto, a informação contida em cada frame do vídeo muda com o tempo e deve-se investigar se existe alguma dinâmica que rege essa mudança.

Seja $\mathbf{h}[k]$ um vetor com medidas de informaçãs da variável aleatória X de um processo estocástico de tempo discreto. A existência de dinâmica no comportamento da informação dessa V.A. está condicionada a encontrarmos uma função f tal que

$$\mathbf{h}[k+1] = f(\mathbf{h}[k], u[k]) \quad k = 0, 1, 2, ...$$

em que k é o instante de observação e u é chamado de entrada. Nesse caso, o vetor $\mathbf{h}[k]$ é um vetor de estados construído a partir de informações existentes na variável aleatória X.

A metodologia utilizada nos testes foi baseada na análise da dinâmica da informação utilizando vídeos como exemplos de ambientes dinâmicos. Conforme visto no capítulo 2, existem diversos descritores na teoria da informação que medem a informação de uma dada variável aleatória e que portanto poderiam ser utilizados para compor o vetor $\mathbf{h}[k]$ nesse trabalho. Por questões de conveniência, consideraremos que $\mathbf{h}[k]$ é unidimensional e elegeremos o potencial de informação ($V_2(.)$) como a nossa medida de informação daqui por diante. Chamaremos o $V_2(.)$ de $V(.)$ para efeito de simplicação.

O procedimento inicial das simulações consistiu em se obter uma sequência com os IPs de cada frame do vídeo. O cálculo do IP foi realizado conforme a equação 2.8, onde a VA X representa os valores RGB de cada pixel. A partir daí, fez-se necessário o uso de alguma técnica capaz de encontrar a dinâmica existente nos dados. Existem inúmeras técnicas para esse fim, a depender das características do sistema. Nessa modelagem inicial, optamos por fazer a identificação e a predição dos sistemas por meio de uma rede NAR (Rede Neural Autorregressiva). A equação 4.1 apresenta a definição para o modelo NAR, onde o valor seguinte do sinal de saída y[k] é dependente de valores anteriores do sinal de saída.

$$y[k] = f(y[k-1], y[k-2], \ldots, y[k-N]) \tag{4.1}$$

Nesse trabalho, o sinal y[k] será representado pelo potencial de informação do frame no instante k. Em outras palavras temos que:

$$\hat{V}[k] = f(\hat{V}[k-1], \hat{V}[k-2], \hat{V}[k-3], \hat{V}[k-4]) \tag{4.2}$$

A figura 4.3 apresenta um esquema da arquitetura da NAR utilizando o IP como sinal de entrada.

Para a simulação da rede NAR fez-se uso da sigmóide (φ) como função de ativação dos neurônios. Considerando que a rede possui q neurônios na camada de entrada e p neurônios na camada escondida, a saída $\hat{V}[k+1]$ está expressa na equação 4.3.

$$\hat{V}[k+1] = \sum_{j=1}^{p} \varphi(\sum_{i=1}^{q} w_{ij} \hat{V}[k+1-i] + b_j) \tag{4.3}$$

onde b_j é o limiar de ativação, w_{ij} é o peso entre os neurônios i e j e

$$\varphi(\upsilon) = \frac{1}{1 + exp(-\upsilon)}$$

Para o treinamento da rede utilizou-se o algoritmo Levenberg-Marquardt e o erro médio quadrático (MSE) como critério de otimização. Seja θ o vetor com os pesos da rede neural, o MSE de θ é definido da seguinte forma:

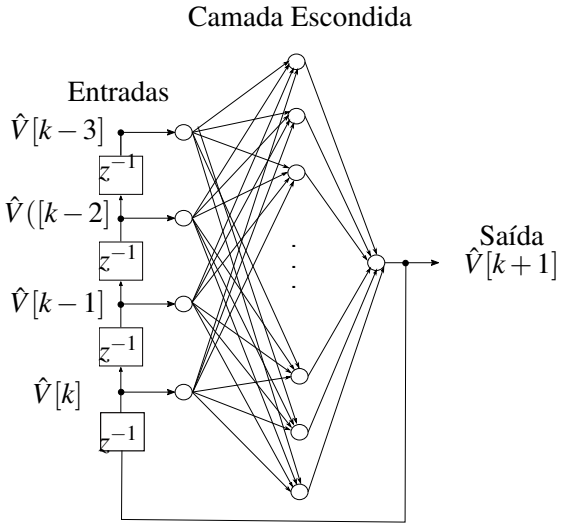


Figura 4.3: Esquema da Rede NAR

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathbf{E}\{(\hat{V}[k] - V[k])^2\}$$

Onde o símbolo **E** denota a operação de valor esperado ou esperança.

Foram realizadas análises da função de autocorrelação do erro (e), com o objetivo de se avaliar a adequação do modelo aos sistemas testados. Considera-se o erro da rede (e), a diferença entre a saída estimada pela rede NAR ($\hat{\mathbf{V}}$) e o seu valor real ($\mathbf{V}$). De acordo com (Box et al. 1994), a função de autocorrelação mede a correlação entre $e[k]$ e $e[k+w]$, onde $w = 0, ..., W$ e $e[k]$ é um processo estocástico. A definição da autocorrelação para o lag w é dada por

$$r_w = \frac{c_w}{c_0}$$

onde c_0 é a variância da amostra da série temporal e

$$c_w = \frac{1}{K-1} \sum_{k=1}^{K-w} (e[k] - \bar{e})(e[k+w] - \bar{e})$$

A saída da NAR, após o treinamento, apresentará uma predição do próximo valor do IP em função de seus valores passados, comprovando a presença de dinâmica na informação do vídeo ao longo do tempo. Os resultados obtidos a partir dos experimentos são apresentados na seção 4.2.

4.2 Experimentos e Resultados - Parte 1

Esta seção apresenta os resultados obtidos em experimentos com três vídeos com características distintas. O primeiro vídeo apresenta frames bastante semelhantes ao longo

do tempo, sem grandes variações de cores; o segundo vídeo é uma animação bem colorida com frames com comportamento bastante aleatórios e o terceiro vídeo é um comercial com efeitos de fadein e fadeout. O comportamento do IP foi analisado em cada um desses vídeos, ressaltando-se as peculiaridades de cada um deles. Além disso, também realizou-se uma análise da existência de dinâmica da informação desses vídeos ao longo do tempo com o auxílio de uma rede neural autorregressiva.

Experimento com o Vídeo 1- IP Constante

O primeiro vídeo analisado trata-se de uma animação com 300 frames, uma resolução de 480x360 e uma taxa de atualização de 25 quadros/s. A figura 4.4 apresenta a imagem de um dos frames do referido vídeo.

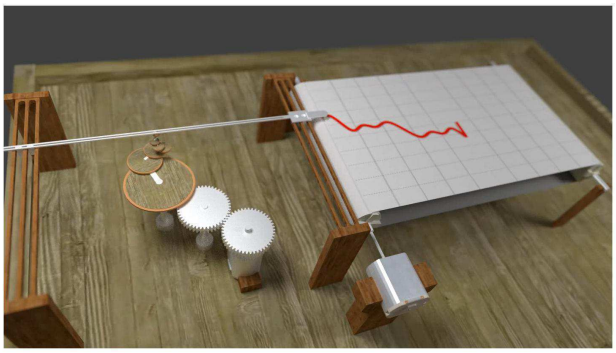


Figura 4.4: Frame do vídeo 1.

Como o vídeo em questão apenas simula o plot da curva vermelha ao longo do tempo, os frames do mesmo são bastante parecidos, sem grandes alterações de cores ao longo do tempo. A figura 4.5 apresenta o comportamento do IP para o vídeo 1.

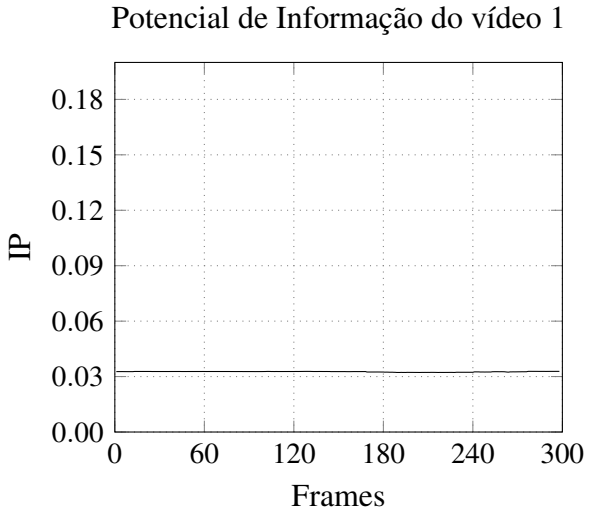


Figura 4.5: Comportamento do IP ao longo do vídeo 1

Observe que os valores do IP se apresentam praticamente constantes ao longo do tempo. Esse comportamento já era esperado, pois, conforme citado anteriormente, o potencial de informação de cada instante de tempo é calculado em função dos valores RGB encontrados em cada pixel do frame, os quais não se alteram ao longo do tempo. A figura 4.6 apresenta uma sequência de frames do vídeo com suas respectivas nuvens RGB.

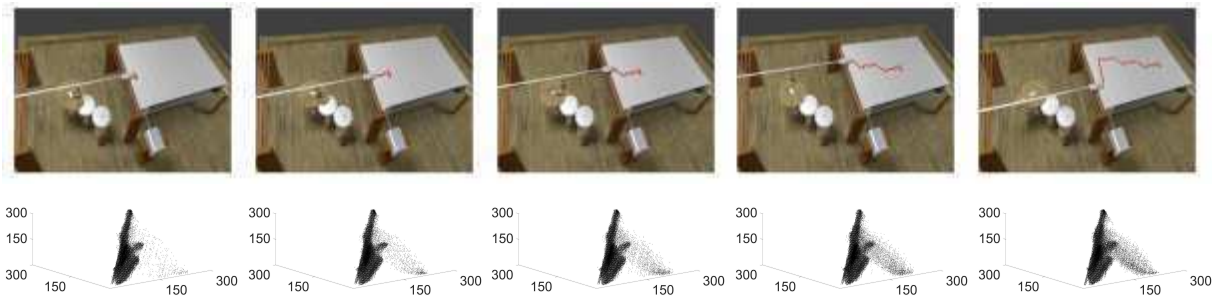


Figura 4.6: Sequência de frames e gráficos RGB do vídeo 1

Note na figura 4.6, que todas as nuvens apresentam formato bastante parecidos. Isso se deve ao fato de que as imagens ao longo do vídeo são praticamente iguais, diferenciando apenas no formato da curva vermelha. Esse padrão de cores repetitivo é o que torna os valores de IP constantes na figura 4.5.

Experimento com o Vídeo 2 - Alterações Bruscas no IP

O segundo vídeo analisado trata-se de uma animação bastante colorida com 3000 frames, uma resolução de 480x360 e uma taxa de atualização de 25 quadros/s. A figura 4.7 apresenta a imagem de um dos frames do referido vídeo.

Figura 4.7: Frame do vídeo 2. Fonte: El Espantapájaros - Estudios de Animacion ICAIC. https://www.youtube.com/watch?v=eKzyalZgJxQ

Conforme citado anteriormente, o potencial de informação de cada instante de tempo é calculado em função dos valores RGB encontrados em cada pixel do frame. A figura 4.8 apresenta o comportamento do IP para o vídeo 2. Observe que os valores do IP

apresentam comportamento aleatório ao longo do tempo e que além disso, permanece em valores abaixo de 0,03 na grande maioria dos frames.

Um fato importante a se notar é a presença de picos nos valores do IP em alguns pontos do vídeo. Esse fato se justifica por uma mudança repentina no padrão de cores apresentados ao longo do vídeo. Na figura 4.8, dois pontos foram destacados por apresentar mudança substancial no potencial de informação. O ponto 1 localiza-se no frame 735, onde acontece uma pequena sequência de frames praticamente pretos, enquanto que o ponto 2 localiza-se no frame 1250 e também apresenta uma sequência de frames mais escuros que a média do vídeo.

As sequências de frames com seus respectivos gráficos de valores RGB, referentes aos pontos 1 e 2, podem ser observadas nas figuras 4.9 e 4.10.

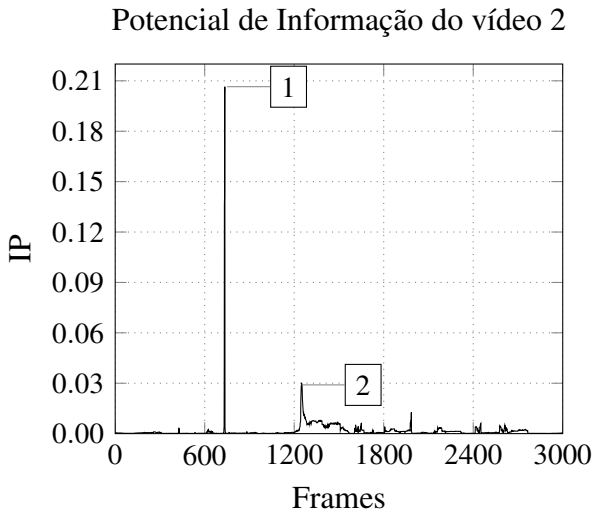


Figura 4.8: Comportamento do IP ao longo do vídeo 2

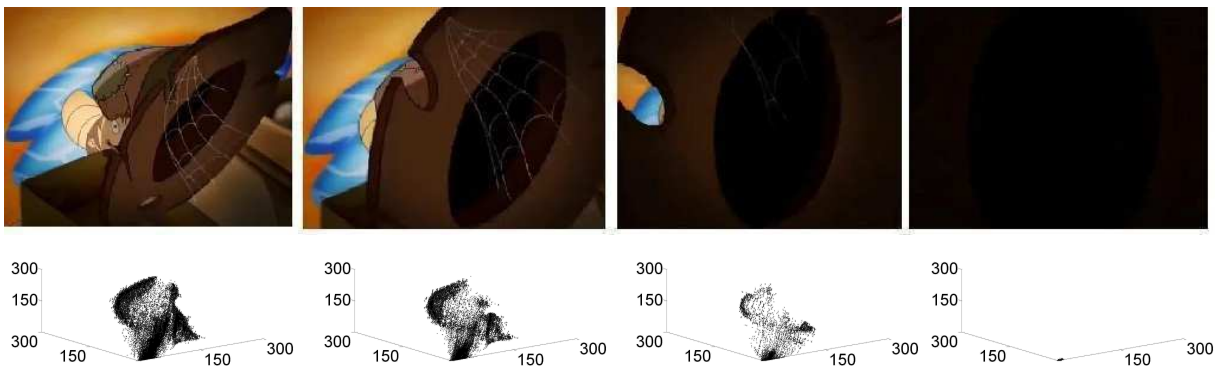


Figura 4.9: Sequência de frames e gráficos RGB no ponto 1

Observe na figura 4.9 que a nuvem RGB referente ao primeiro frame é bastante esparsa, refletindo o colorido do mesmo. Porém, à medida que o vídeo apresenta frames

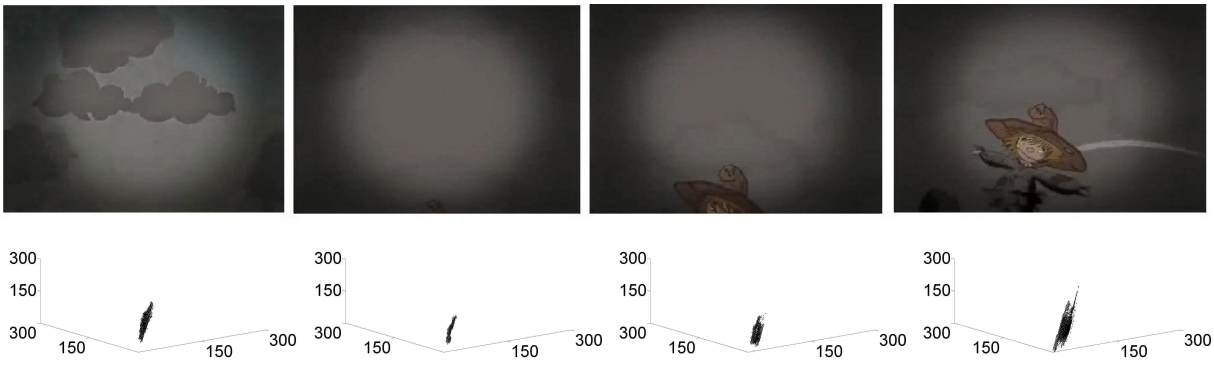


Figura 4.10: Sequência de frames e gráficos RGB no ponto 2

mais escuros, a nuvem RGB tende a diminuir, ficando com seus pontos bem concentrados próximo à origem dos eixos. É neste momento que acontece o pico representado pelo ponto 1, já que o IP de uma variável aleatória é inversamente proporcional à variância dos dados.

Note na figura 4.10, que todas as nuvens são compactas com formato bastante parecidos. Isso se justifica, pelo padrão de cores acinzentadas que predomina em toda a sequência apresentada na figura 4.10. Esses tons escuros nos frames, tornam o valor do IP mais altos que a média do vídeo, pois a informação está sendo calculada em função dos valores RGB de cada pixel.

A sequência de IPs do vídeo, apresentada na figura 4.8 foi utilizada no treinamento da rede NAR e os resultados dos testes são apresentados na figura 4.11. A partir dos testes, podemos perceber que a rede neural conseguiu realizar a predição dos dados de forma bastante aceitável, apresentando uma soma dos erros médios quadráticos de $9.61x10^{-5}$.

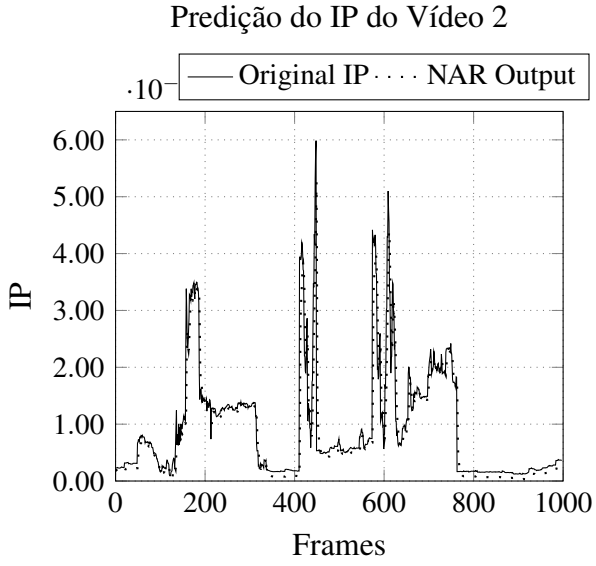


Figura 4.11: Resultado dos testes da NAR - vídeo 2

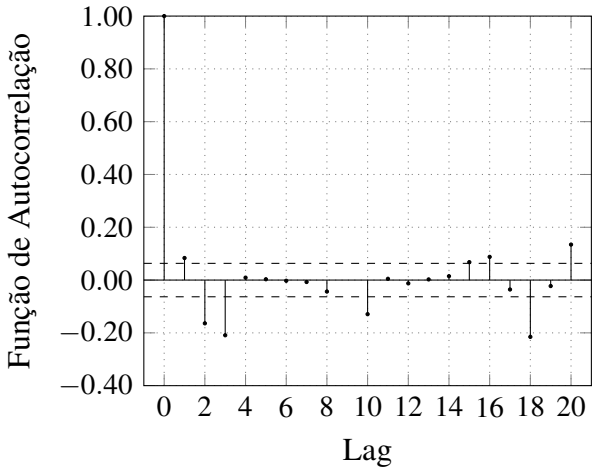


Figura 4.12: Função de Autocorrelação - Vídeo 2

A partir da função de autocorrelação do erro, apresentada na figura 4.12, pode-se notar que apesar da rede ter sido capaz de identificar alguma dinâmica presente na informação do sistema, ainda restou uma dinâmica dos dados que não foi completamente capturada. Isso significa que os parâmetros de configuração da rede ainda podem ser melhorados, a fim de obtermos uma maior eficiência.

A confirmação da presença de dinâmica nos dados se mostra como um resultado bastante interessante, pois nos permite utilizá-la como informação em processos de compressão ou de análise de vídeos.

Experimento com o Vídeo 3 - Efeitos Fadeout e Fadein

O terceiro vídeo utilizado nos testes, trata-se de um vídeo comercial com 420 frames, uma resolução de 1280x720 e uma taxa de 9 quadros/s. A figura 4.13 apresenta a imagem de um dos frames do vídeo 3.

Figura 4.13: Frame do vídeo 3 - Fonte: Comerciais Engraçados Parte 13 - Canal Chuva na Nuca. https://www.youtube.com/watch?v=Z3uFS wYpk0

O comportamento do IP de cada frame ao longo do tempo pode ser observado na figura 4.14. Para o vídeo 3, pode-se observar uma variação maior nos valores do IP ao longo do tempo quando comparado com o vídeo testado anteriormente. Note que são apresentados valores de IP bem mais altos que a média no início e no fim do vídeo, onde existem efeitos de fadeout e fadein, respectivamente, apresentando frames quase totalmente pretos em ambos os casos.

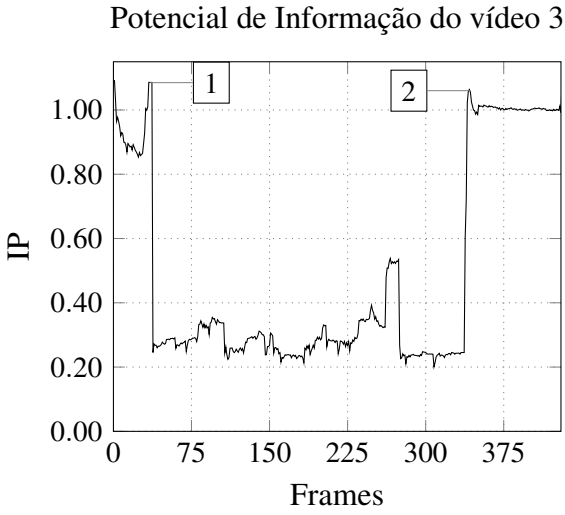


Figura 4.14: Comportamento do IP ao longo do vídeo 3

Nas figuras 4.15 e 4.16 podemos observar imagens dos frames com os efeitos fadeout e fadein no vídeo 3, bem como seus respectivos gráficos de valores RGB, representados pelos pontos 1 e 2 na figura 4.14.

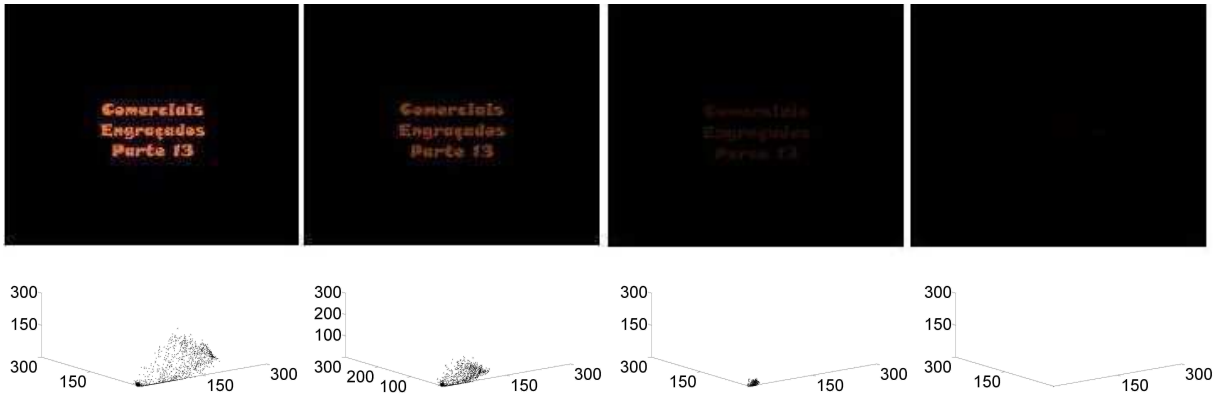


Figura 4.15: Sequência de Frames e Gráficos RGB - Ponto 1 : Efeito Fadeout

A partir da figura 4.15 pode-se observar a nuvem RGB dos pixels se concentrando próximo à origem dos eixos à medida que o frame se torna mais escuro em função do efeito Fadeout, ocasionando o pico no valor do IP representado pelo ponto 1 na figura 4.14. Já na figura 4.16, podemos observar o efeito contrário, onde a nuvem RGB inicialmente concentrada, se expande gradualmente à medida que o efeito fadein acontece.

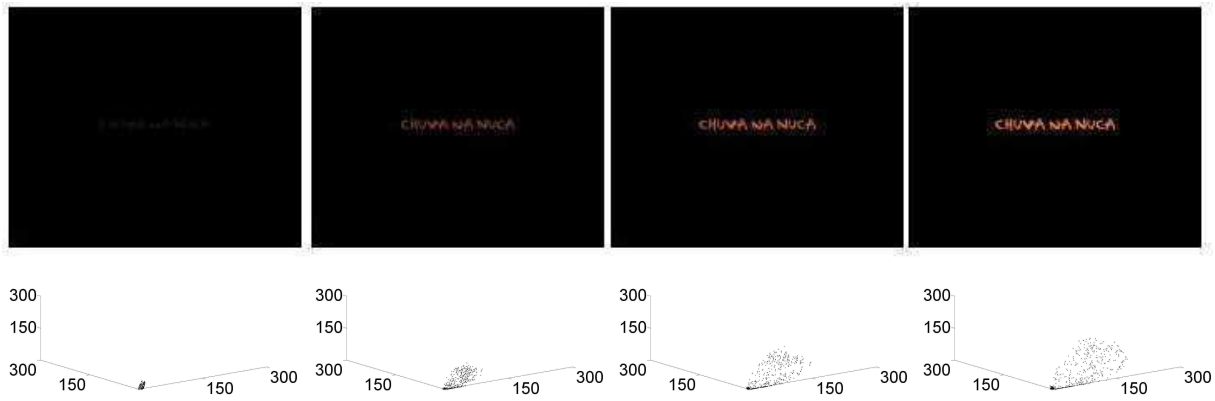


Figura 4.16: Sequência de Frames e Gráficos RGB - Ponto 2 : Efeito Fadein

Os resultados dos testes com a rede NAR são apresentados na figura 4.17. A partir dos testes, podemos perceber que a rede neural se mostrou eficaz na predição dos dados, apresentando uma soma dos erros médios quadráticos de $3.41x10^{-1}$. De forma análoga ao segundo vídeo, esse gráfico nos leva a novamente concluir que também existe uma dinâmica presente na informação do terceiro vídeo, a qual a rede NAR foi capaz de identificá-la.

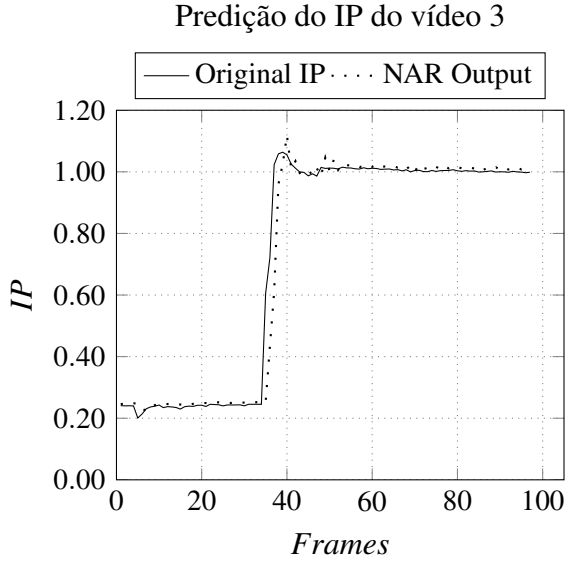


Figura 4.17: Resultado dos testes da NAR - vídeo 3

A figura 4.18 apresenta a função de autocorrelação do resíduo entre o IP calculado e o estimado pela rede. Apesar da rede ter identificado uma dinâmica nos dados e realizado uma predição adequada, pode-se perceber por meio dos lags iniciais da figura 4.18 que a configuração da rede não foi capaz de extrair toda a dinâmica presente nos dados, sugerindo uma adequação em seus parâmetros para obtenção de melhores resultados.

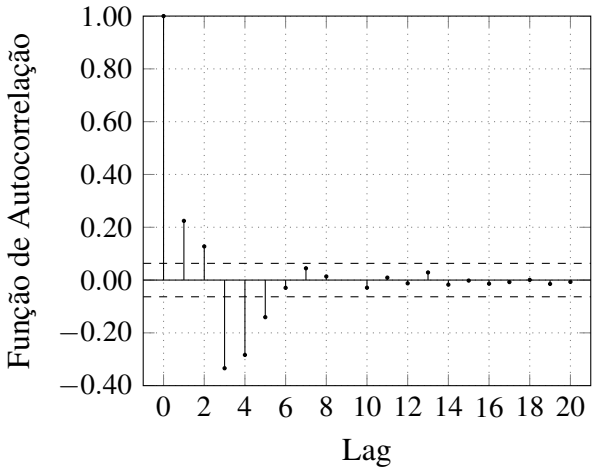


Figura 4.18: Função de Autocorrelação - Vídeo 3

Os resultados desses experimentos se mostraram bastante interessantes, pois mostrou que a rede NAR foi capaz de identificar e medir a dinâmica da informação presente nos dados de forma adequada. Além disso, se apresenta de forma promissora, já que nos abre um leque de possibilidades de aplicações da área de teoria da informação dentro do contexto dos sistemas dinâmicos. Dentre as diversas áreas possíveis para aplicações, destacam-se os sistemas de monitoramento por imagens, onde alertas são disparados diante de mudanças significativas no conteúdo das informações. Podemos ainda sugerir o uso da informação como uma variável adicional na modelagem dos sistemas dinâmicos por meio das técnicas tradicionais.

4.3 Modelo Estados de Informação

Diante da comprovação da presença de dinâmica na informação, essa seção apresenta o conceito dos Estados de Informação, que representa uma estrutura a ser utilizada na modelagem de processos dinâmicos baseada na sua informação. Essa modelagem consiste de uma representação espaço-estados na qual as variáveis de estado são medidas de informação do sistema.

A representação espaço-estados para sistemas de tempo discreto também conhecida como modelo espaço de estados, geralmente tem a seguinte forma:

$$\mathbf{z}[k+1] = G(\mathbf{z}[k], \mathbf{u}[k], \varepsilon[k]) \tag{4.4}$$

$$y[k] = H(\mathbf{z}[k], \mathbf{u}[k], \delta[k]) \tag{4.5}$$

em que $\mathbf{z}[k]$ é o vetor de variáveis de estado, $\mathbf{u}[k]$ é uma entrada opcional ou sinal de entrada, $y[k]$ é a observação, G é o modelo de transição de estados, H é o modelo de observação, $\varepsilon[k]$ é o ruído do sistema no tempo k e $\delta[k]$ é o ruído da observação no tempo k.

As variáveis de estado modeladas por $\mathbf{z}[k]$ devem descrever completamente o comportamento do processo dinâmico. Vários modelos de variáveis de estado podem ser obtidos para um mesmo sistema, a depender da escolha dessas variáveis. Nesse modelo, as variáveis de estado do processo dinâmico são modeladas a partir da informação contida em seus próprios dados, por meio do uso dos descritores da teoria da informação.

Seja $\mathbf{I}[k]$ um vetor com medidas de informação de uma dada variável aleatória de um processo estocástico no instante k. Chamaremos $\mathbf{I}[k]$ de estado de informação. A ideia desse trabalho é que a variável de estado do sistema ($\mathbf{z}[k]$) seja representada por um estado de informação ($\mathbf{I}[k]$).

Das equações 4.4 e 4.5, temos que a existência de dinâmica no comportamento da informação dessa V.A. está condicionada a encontrarmos as funções G e H, tal que

$$\mathbf{I}[k+1] = G(\mathbf{I}[k], \mathbf{u}[k], \varepsilon[k]) \tag{4.6}$$

$$y[k] = H(\mathbf{I}[k], \mathbf{u}[k], \delta[k]) \tag{4.7}$$

O processo de estimação dos estados de um sistema dinâmico na presença de ruídos pode ser eficientemente realizado por meio do filtro de Kalman (FK). Nesse filtro, a estimação da variável não observável, denominada "variável de estado", é realizada através da observação da variável denominada "variável de observação".

O modelo estados de informação (MEI) é apresentado nesse trabalho como uma nova forma de representar processos dinâmicos em função de suas medidas de informação.

Conceitualmente, o MEI é um modelo de espaço de estados, conforme descrito nas equações 4.6 e 4.7, onde o vetor de estados $\mathbf{I}[k]$ é um vetor de quantidades de informação do sistema dinâmico e os modelos de transição e observação dos estados, G e H, são funções dessas quantidades de informação. A saída $y[k]$, representa uma estimativa da variável observada do sistema.

Um caso especial do modelo de estados de informação é quando todas as variáveis observadas e não observadas apresentam distribuição normal e os modelos de transição e observação são lineares. Nesse caso assume-se que:

- O modelo de transição G é uma função linear do vetor de quantidades de informação $\mathbf{I}[k]$:

$$\mathbf{I}[k+1] = \mathbf{A}\,\mathbf{I}[k] + \mathbf{B}\,\mathbf{u}[k] + \varepsilon[k] \tag{4.8}$$

- O modelo de observação H é uma combinação $\mathbf{C}$ das quantidades de informação que compõem o vetor $\mathbf{I}[k]$:

$$y[k] = \mathbf{C}\,\mathbf{I}[k] + \mathbf{D}\,\mathbf{u}[k] + \delta[k] \tag{4.9}$$

- Os ruídos do sistema e da observação são gaussianos:

$$\varepsilon[k] \sim N(\mathbf{0}, \mathbf{Q}[k])$$

$$\delta[k] \sim N(\mathbf{0}, \mathbf{R}[k])$$

Desconsiderando as entradas $\mathbf{u}[k]$ e os ruídos $\varepsilon[k]$ e $\delta[k]$ nas equações 4.8 e 4.9 , nosso modelo é dado por:

$$\mathbf{I}[k+1] = \mathbf{A}\,\mathbf{I}[k] \tag{4.10}$$

$$y[k] = \mathbf{C}\,\mathbf{I}[k] \tag{4.11}$$

Vale lembrar que $\mathbf{I}[k]$ é um vetor de q quantidades de informação do sistema no instante k, o qual poderá ser composto por quaisquer descritores da teoria da informação.

$$\mathbf{I} = \begin{pmatrix} I_1[k] \\ I_2[k] \\ \vdots \\ I_q[k] \end{pmatrix} \tag{4.12}$$

As matrizes $\mathbf{A}$ e $\mathbf{C}$, representadas nas equações 4.10 e 4.11, são obtidas a partir de um modelo linear qualquer das quantidades de informação do sistema $\mathbf{I}[k]$. A fim de exemplificação, utilizamos um modelo autorregressivo (AR) de ordem q de $\mathbf{I}[k]$ para obter essas matrizes. Nesse caso, ao invés de utilizarmos diferentes medidas de informação para compor nosso estado de informação $\mathbf{I}$[k], faremos uso dos valores de um único descritor em instantes passados, conforme mostra a equação 4.13:

$$\mathbf{I} = \begin{pmatrix} I_1[k] \\ I_2[k] \\ \vdots \\ I_q[k] \end{pmatrix} = \begin{pmatrix} I[k-q] \\ I[k-q+1] \\ \vdots \\ I[k-1] \end{pmatrix} \tag{4.13}$$

O modelo AR(q) de $\mathbf{I}[k]$ é apresentado nas equações 4.14 e 4.15.

$$I[k] = \sum_{i=k-q}^{k-1} a[i]I[i] \tag{4.14}$$

$$I[k] = a[k-q]I[k-q] + \cdots + a[k-1]I[k-1] \tag{4.15}$$

Para se obter uma representação por espaço de estados do modelo AR representado na equação 4.15 é necessário transformá-lo para a forma de avanço, conforme mostra a equação 4.16.

$$I[k+q] = a[k+q-1]I[k+q-1] + \cdots + a[k]I[k] \tag{4.16}$$

Colocando o nosso estado de informação, representado na equação 4.13, na forma de avanço, temos:

$$\begin{pmatrix} I_1[k] \\ I_2[k] \\ \vdots \\ I_q[k] \end{pmatrix} = \begin{pmatrix} I[k] \\ I[k+1] \\ \vdots \\ I[k+q-1] \end{pmatrix} \tag{4.17}$$

No instante k+1 temos o seguinte vetor de estados de informação:

$$\begin{pmatrix} I_1[k+1] \\ I_2[k+1] \\ \vdots \\ I_q[k+1] \end{pmatrix} = \begin{pmatrix} I[k+1] \\ I[k+2] \\ \vdots \\ I[k+q] \end{pmatrix} \tag{4.18}$$

Das equações 4.16 e 4.17, temos que:

$$I[k+q] = a[k+q-1]I_q[k] + \cdots + a[k]I_1[k] \tag{4.19}$$

Dessa forma:

$$\begin{pmatrix} I_1[k+1] \\ I_2[k+1] \\ \vdots \\ I_q[k+1] \end{pmatrix} = \begin{pmatrix} I_2[k] \\ I_3[k] \\ \vdots \\ a[k+q-1]I_q[k] + \cdots + a[k]I_1[k] \end{pmatrix} \tag{4.20}$$

Representando a equação 4.20 na forma matricial, tem-se:

$$\begin{pmatrix} I_1[k+1] \\ I_2[k+1] \\ \vdots \\ I_q[k+1] \end{pmatrix} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ a[k] & a[k+1] & \cdots & a[k+q-1] \end{pmatrix} \begin{pmatrix} I_1[k] \\ I_2[k] \\ \vdots \\ I_q[k] \end{pmatrix} \tag{4.21}$$

Se fizermos a variável de saída do modelo de observação, $y[k]$, como a quantidade de informação I no instante k, temos:

$$y[k] = I[k] = I_1[k] \tag{4.22}$$

$$y[k] = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} I_1[k] \\ I_2[k] \\ \vdots \\ I_q[k] \end{pmatrix} \tag{4.23}$$

Logo a representação espaço de estados para o modelo autorregressivo de $I[k]$ apresentado na equação 4.15 se dá da seguinte forma:

$$\mathbf{I}[k+1] = \mathbf{A}\,\mathbf{I}[k] \tag{4.24}$$

$$y[k] = \mathbf{C}\,\mathbf{I}[k] \tag{4.25}$$

Onde:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & 1 \\ a[k] & a[k+1] & a[k+2] & \cdots & a[k+q-1] \end{pmatrix}$$

$$, \ \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

e a[k] são os coeficientes do modelo AR da medida de informação do sistema.

4.4 Experimentos e Resultados - Parte 2

Nesta seção demonstramos uma aplicação do MEI na avaliação da qualidade de vídeos ruidosos por meio de sua informação. O objetivo é avaliar a qualidade dos vídeos sem fazer uso do vídeo original. Essa aplicação parte da premissa de que quanto mais ruidoso esteja o vídeo, pior será sua qualidade. Nesse caso, o experimento consistiu em avaliar os resíduos do IP dos vídeos após uso do filtro de Kalman utilizando o modelo estados de informação.

Existem diversos descritores na teoria da informação que medem a informação de uma dada variável aleatória e que, portanto, poderiam ser utilizados para compor o vetor de estados de informação nesse trabalho. Por questões de conveniência, utilizaremos o modelo estado de informação linear com função autoregressiva e elegeremos o potencial de informação ($V_2(.)$) como a nossa medida de informação daqui por diante. Chamaremos o $V_2(.)$ de $V(.)$ para efeito de simplicação.

O procedimento inicial para o experimento de avaliação da qualidade dos vídeos ruidosos, consistiu em se obter o comportamento do potencial de informação de cada frame ao longo do vídeo ruidoso. Esse IP foi calculado em função do padrão RGB dos pixels de cada frame, conforme a equação 2.8. A sequência de IPs obtida dos frames do vídeo serviu para gerar um modelo AR do sistema com ordem 5. Os coeficientes do modelo AR do IP foram obtidos com o auxílio da função **ar** do Matlab e os mesmos foram utilizados para se obter o modelo na forma de espaço de estados, utilizando o conceito de estados de informação.

Após a obtenção do modelo espaço de estados baseado nos estados de Informação (MEI), utilizou-se o filtro de Kalman para se obter o resíduo do IP dos vídeos ruidosos e poder então avaliar a qualidade dos mesmos. Entende-se por resíduo do IP, a diferença entre a medida do IP calculado e a medida do IP estimado pelo filtro de Kalman.

Foram realizados dois estudos de caso: O primeiro comparou a qualidade dos vídeos com adição de diferentes níveis de ruído gaussiano e o segundo avaliou os vídeos com introdução de ruídos impulsivos também em níveis diferentes. Os testes foram realizados em 3 vídeos com características distintas. Esses experimentos foram realizados no software Matlab R2013a versão 8.0.604.

4.4.1 Estudo de caso 1: Avaliação da qualidade de vídeos submetidos a diferentes níveis de ruído gaussiano

Nesse teste, para cada vídeo, ruídos gaussianos em níveis diferentes foram introduzidos aleatoriamente pixel a pixel em todos os frames do mesmo. Isto nos resultou em 2 vídeos com intensidades diferentes de ruído para cada vídeo original: um vídeo com pouco ruído introduzido e um vídeo com bastante ruído introduzido. A figura 4.19 apresenta frames de um dos vídeos avaliados com intensidades diferentes de ruído adicionado.

Figura 4.19: (a)Vídeo com baixo nível de ruído. (b)Vídeo com alto nível de ruído.

As figuras 4.20, 4.21 e 4.22 apresentam os resíduos dos IPs para os vídeos analisados após introdução do ruído gaussiano. A tabela 4.1 apresenta um resumo das médias dos resíduos para os 3 vídeos analisados.

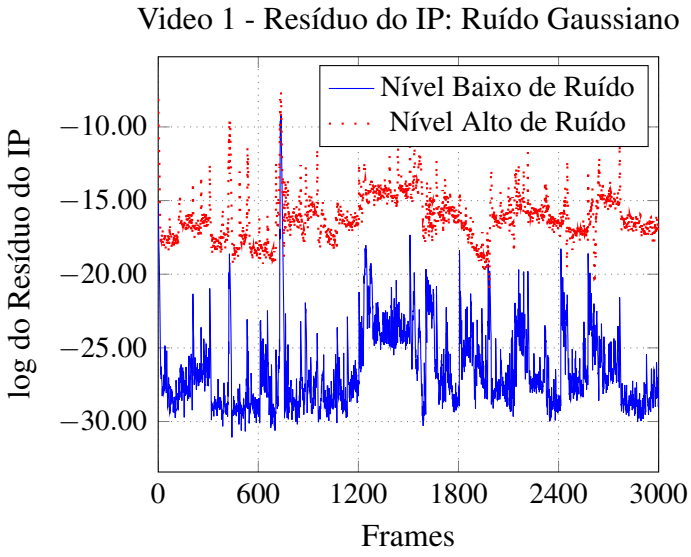


Figura 4.20: Resíduo do IP após Filtro de Kalman: Vídeo 1 com Ruído Gaussiano

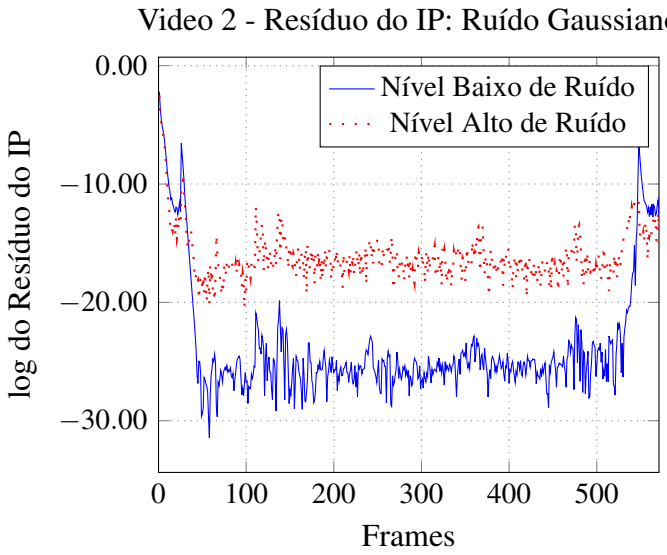


Figura 4.21: Resíduo do IP após Filtro de Kalman: Vídeo 2 com Ruído Gaussiano

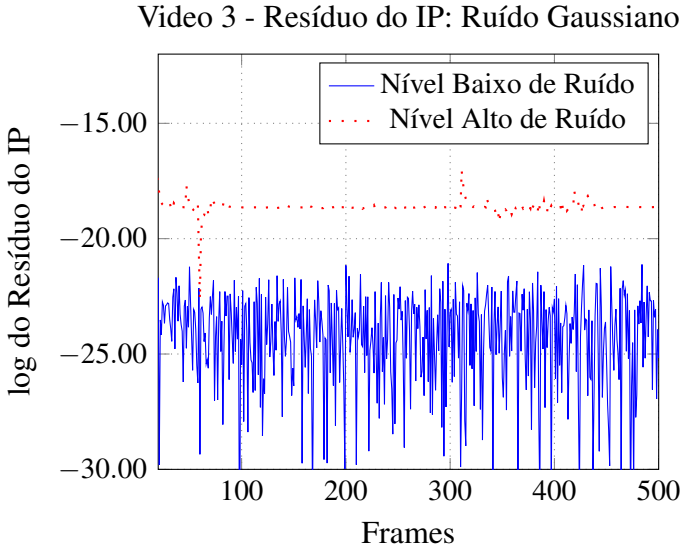


Figura 4.22: Resíduo do IP após Filtro de Kalman: Vídeo 3 com Ruído Gaussiano

Tabela 4.1: Média dos Resíduos dos IP's - Ruído Gaussiano

Nível do Ruído	Video 1	Video 2	Video 3
Nível Baixo de Ruído	$1.382E10^{-6}$	$3.0438E10^{-11}$	$9.1796E10^{-11}$
Nível Alto de Ruído	$1.942E10^{-6}$	$1.6564E10^{-07}$	$8.3207E10^{-09}$

Note que o valor da média do resíduo do IP do vídeo com muito ruído é sempre bem maior que o valor médio do resíduo do vídeo com pouco ruído introduzido. Nesse

caso, podemos inferir que o vídeo com melhor qualidade apresenta um resíduo de IP menor após filtragem quando comparado com o mesmo vídeo com qualidade inferior, confirmando nossa premissa.

4.4.2 Estudo de caso 2: Avaliação da qualidade do vídeo submetido a diferentes níveis de ruído impulsivo

Para esse teste, foram introduzidos, aleatoriamente, ruídos impulsivos em um certo percentual dos pixels de cada frame. Nesse trabalho, os ruídos impulsivos foram gerados a partir da substituição de alguns pixels do frame por pixels totalmente pretos. Nesse caso, foram comparadas três intensidades diferentes de ruído introduzido, a saber, 6% dos pixels, 10% dos pixels e 40% dos pixels ruidosos.

As figuras 4.23 4.24 e 4.25 apresentam o resíduo dos IPs dos vídeos após introdução do ruído em três vídeos distintos. A tabela 4.2 apresenta um resumo das médias dos resíduos para os 3 vídeos analisados.

É possível notar nas figuras 4.23,4.24 e 4.25, e na tabela 4.2 que, quanto maior é o percentual de ruído adicionado nos vídeos, maior é o seu resíduo de IP. Dessa forma, torna-se possível avaliar a qualidade de vídeos por meio de seu potencial de informação sem a necessidade de se ter o vídeo original, onde vídeos com melhor qualidade (menor incidência de ruídos) apresentarão menor valor residual quando comparados à vídeos com qualidade inferior (maior incidência de ruídos).

Em todas as simulações verificou-se que os resíduos gerados por vídeos com níveis mais altos de ruídos foram sempre maiores do que em vídeos com níveis inferiores de ruído. Além disso, foram obtidos bons resultados, o que motiva uma investigação maior na técnica proposta. Vale ressaltar que o modelo estados de informação possui potencial de aplicação em diferentes áreas, tais como controle de sistemas dinâmicos, predição e clusterização dinâmica.

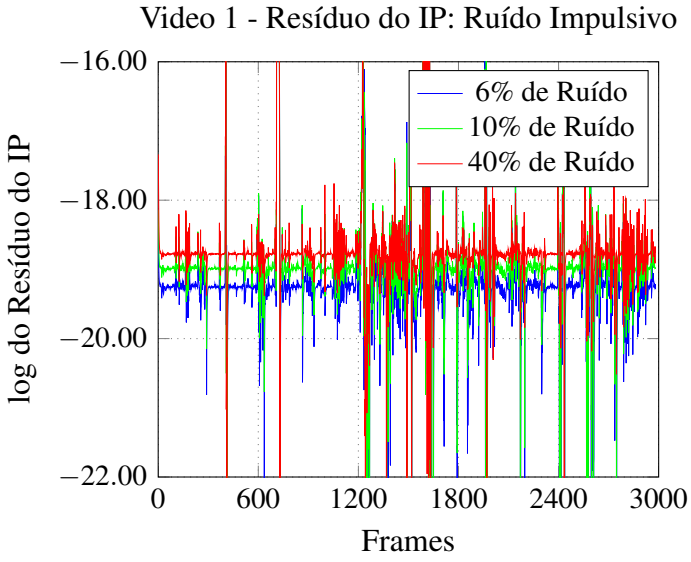


Figura 4.23: Resíduo do IP após Filtro de Kalman: Vídeo 1 com Ruído Impulsivo

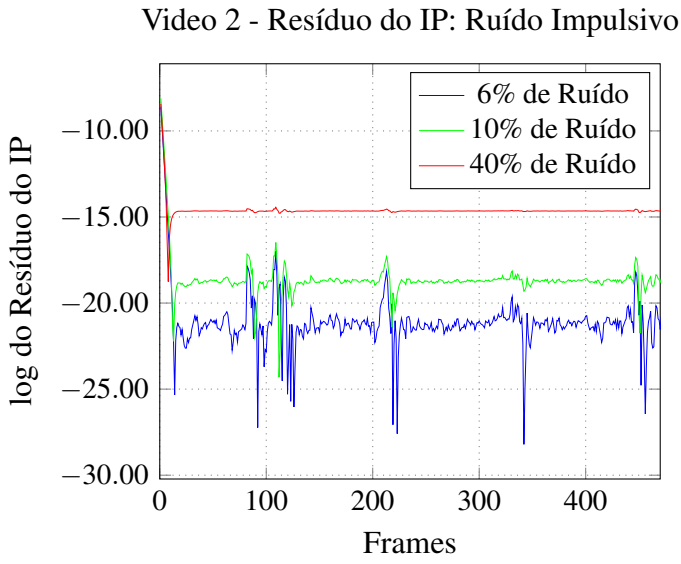


Figura 4.24: Resíduo do IP após Filtro de Kalman: Vídeo 2 com Ruído Impulsivo

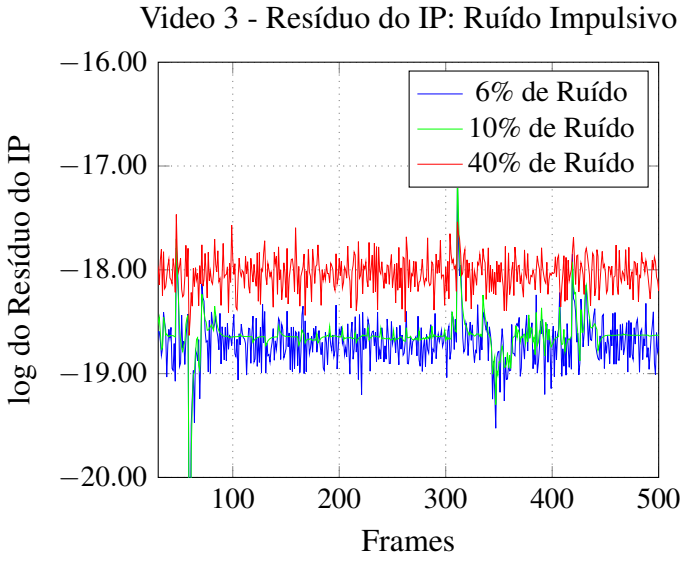


Figura 4.25: Resíduo do IP após Filtro de Kalman: Vídeo 3 com Ruído Impulsivo

Tabela 4.2: Média dos Resíduos dos IP's - Ruído Impulsivo

Nível do Ruído	Video 1	Video 2	Video 3
6%	$1.320E10^{-5}$	$1.1341E10^{-09}$	$7.8865E10^{-09}$
10%	$1.472E10^{-5}$	$8.0298E10^{-09}$	$8.3207E10^{-09}$
40%	$3.344E10^{-5}$	$4.3341E10^{-07}$	$1.5112E10^{-08}$

Capítulo 5

Aplicação em Clusters Dinâmicos

Diante dos resultados obtidos na seção 4.1, onde observou-se a presença de dinâmica na informação de processos dinâmicos, esta seção se dedica ao estudo do uso de medidas de teoria de informação no contexto dos clusters dinâmicos.

O termo clustering dinâmico refere-se ao processo de agrupamento de dados dinâmicos, ou seja, conjuntos de dados que tem suas características alteradas ao longo do tempo. Essa dinâmica no conjunto de dados pode ser caracterizada pela chegada de novos dados ao conjunto constatemente, como em stream de dados, ou pela alteração dos valores dos atributos dos dados ao longo do tempo em conjuntos com tamanho constante. Em ambos os casos, temos uma dinâmica incorporada no problema e precisamos recorrer ao uso dos algoritmos de clustering dinâmico.

No capítulo 3 foram apresentadas algumas técnicas utilizadas atualmente no contexto dos clusters dinâmicos. A maioria desses algoritmos trabalham de forma incremental, sendo adequados para o uso em problemas com stream de dados, os quais recebem os dados recém-chegados e os encaixam na estrutura de clusters já formados anteriormente. Independente da técnica utilizada, após os clusters terem se formado, os mesmos devem ser avaliados a fim de se descobrir se há a necessidade de reajuste na nova estrutura. Esse reajuste de estrutura inclui as operações de junção e separação entre clusters e é realizado sob demanda de acordo com o monitoramento de medidas de proximidade tradicionais e/ou critérios geométricos. Nesta seção, apresentamos uma metodologia para uso de medidas de proximidade baseadas na teoria da informação em substituição às tradicionais medidas de proximidade como critério para as operações de junção e separação.

Vale salientar que a proposta desse trabalho não é apresentar um novo algoritmo para clustering dinâmico, mas sim apresentar um novo critério para detecção de junção e separação entre clusters em processos dinâmicos. A figura 5.1 apresenta um esquema do fluxo dos dados quando utilizamos o critério apresentado nesse trabalho para detecção de alteração na estrutura dos clusters. O esquema apresentado em 5.1 trata de processos dinâmicos que mantém o tamanho do conjunto de dados constante, alterando-se os valores dos atributos ao longo do tempo. Caso deseje-se implementar esse critério em problemas que envolvem stream de dados, o algoritmo deverá seguir o esquema proposto na figura 5.2.

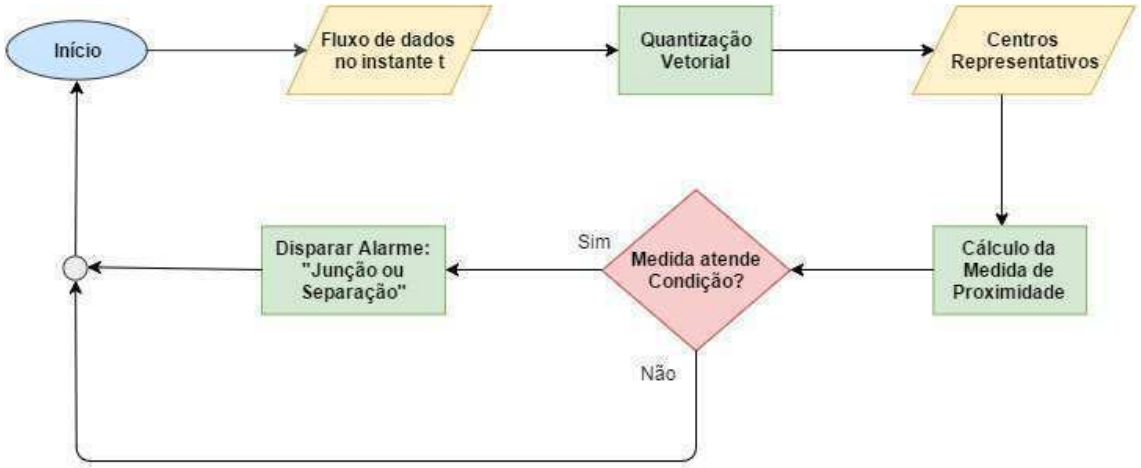


Figura 5.1: Fluxo de dados para Detecção de Junção e Separação entre Clusters em Conjuntos de Dados com Tamanho Constante

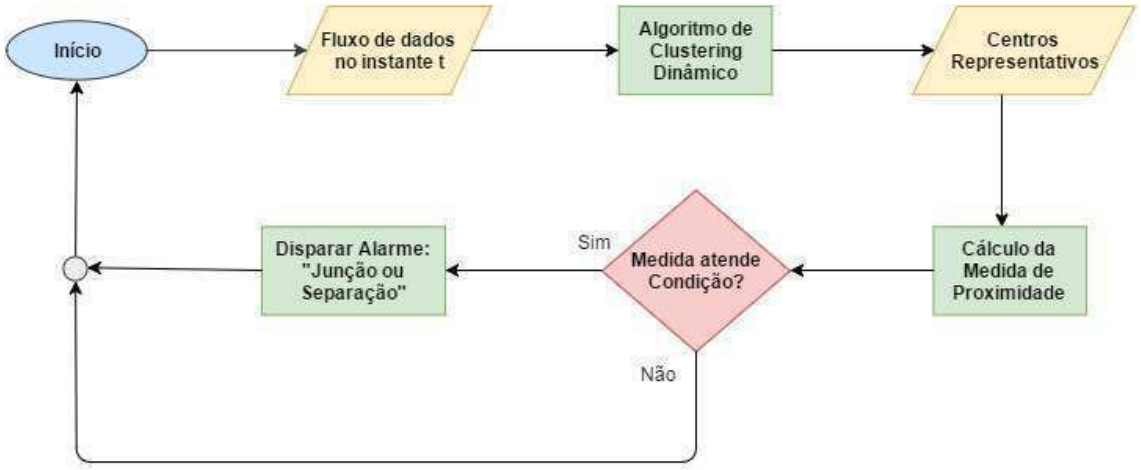


Figura 5.2: Fluxo de dados para Detecção de Junção e Separação entre Clusters em Stream de Dados

5.1 Junção e Separação entre Clusters

Um problema específico no processo de clustering dinâmico acontece quando duas ou mais nuvens de dados locais (cada uma modelada por um cluster) estão se movendo juntas ou são separadas dentro de um cluster. Em tais casos, os clusters devem ser dinamicamente mesclados ou divididos a fim de manter a alta qualidade das partições dos clusters que seguem a distribuição natural das nuvens de dados.

O primeiro caso surge sempre que dois clusters parecem distintos no início do fluxo de dados, mas podem se mover juntos devido a amostras de dados que preenchem as lacunas entre os mesmos. Esse efeito é chamado de merge ou junção de cluster. A Figura

5.3 mostra tal ocorrência, (a) demonstrando a partição (três clusters distintos) após o carregamento do bloco de dados inicial, (b) demonstrando a partição (dois clusters que se moveram juntos) devido ao novo bloco de dados.

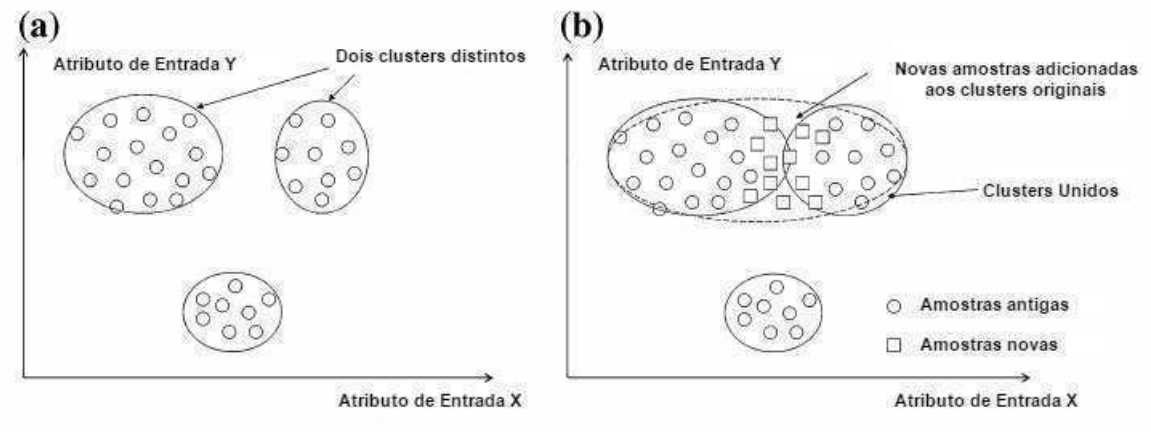


Figura 5.3: Merge: Exemplo de Fusão entre Clusters. Fonte: A dynamic split-and-merge approach for evolving cluster models, Edwin Lughofer. Evolving Systems, 2012.

O segundo caso surge sempre que um cluster em uma região do espaço de dados parece apropriado no início, mas subsequentemente acaba por conter duas nuvens de dados distintas. Esse efeito é chamado de split ou separação de cluster.

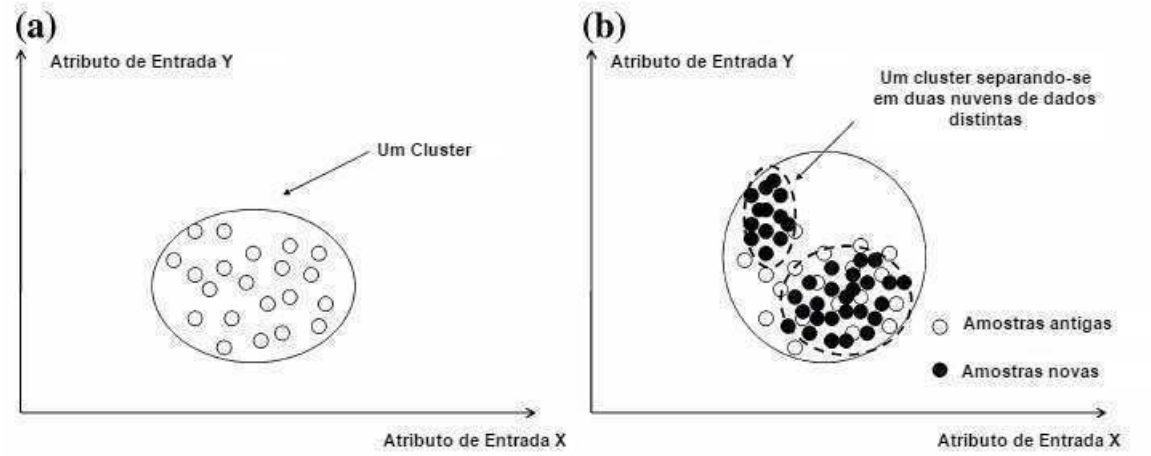


Figura 5.4: Split: Exemplo de Divisão entre Clusters. Fonte: A dynamic split-and-merge approach for evolving cluster models, Edwin Lughofer. Evolving Systems, 2012.

Os critérios de separação e fusão precisam ser avaliados apenas para os clusters já atualizados, ou seja, após a etapa de aprendizado incremental no caso de stream de dados. Nesse trabalho, esses critérios são independentes do algoritmo de atualização de cluster selecionado, porque os critérios apenas identificam mudança na estrutura das partições entre os clusters, independentemente de como os clusters foram movidos, redefinidos, evoluídos ou expandidos.

Segundo um critério geométrico, a condição de mesclagem entre um cluster atualizado e qualquer outro cluster na partição pode ser deduzido a partir da consideração de que os dois clusters são suficientemente próximos uns aos outros, quando seu contorno forma uma estrutura uni-modal homogênea. Sempre que os dois clusters são esféricos, a condição para que dois grupos se sobreponham é dado por:

$$d(C_1, C_2) <= r_1 + r_2 \tag{5.1}$$

onde $d(C_1, C_2)$ denota a distância, calculada por alguma medida de proximidade, entre o cluster atualizado e outro cluster na partição e r_i denota o raio do cluster i.

Para os casos em que os clusters não possuem formato esférico, o calculo deve ser adaptado. Além disso, nem sempre que os clusters se tocam apresentam-se homogêneos o suficiente para que uma operação de junção aconteça. Por esse motivo, neste trabalho ao invés do critério geométrico, utiliza-se medidas de proximidade baseada na teoria de informação afim de se identificar junções e divisões entre clusters.

A ideia desse trabalho é apresentar um novo critério a ser utilizado nos algoritmos de junção e separação entre clusters, o qual pode ser conectado com qualquer algoritmo de clustering dinâmico que possa fornecer pontos representativos do conjunto de dados.

5.2 O Conjunto de Dados

Para realizar os experimentos, inicialmente criou-se um conjunto de dados experimental que tem suas características alteradas ao longo do tempo com o intuito de simular um processo dinâmico real. Nesse caso, o conjunto criado simula uma aplicação que tem quantidade de dados constante porém tem sua dinâmica alterada com o passar do tempo.

O primeiro conjunto de testes foi construido a partir de pontos gerados sob distribuições gaussianas com diferentes médias e variâncias, onde cada gaussiana representa um cluster distinto. Ao longo do tempo, essas gaussianas tem seus parâmetros sistematicamente alterados, de forma que alguns clusters se sobreponham ou precisem se dividir em dois novos clusters. A figura 5.5 apresenta o comportamento desse primeiro conjunto de teste ao longo do tempo.

Observe na figura 5.5 que no instante T=3, o cluster que era único, se dividiu em dois novos clusters que continuam a se afastar com o tempo. No instante T=15, um dos clusters se divide novamente, totalizando 3 clusters no conjunto de dados. No instante T=50 dois clusters se juntam, fazendo com que tenhamos novamente apenas 2 clusters. Esse é um exemplo simples, mas que representa bem um processo com dados dinâmicos.

Em resumo, a metodologia desse estudo consiste em comparar o comportamento de algumas de medidas de proximidade (tradicionais e de teoria da informação) durante a evolução dos dados no tempo. A ideia é calcular o valor de cada medida para cada instante de tempo e analisar seu comportamento afim de que possam ser utilizadas com eficiência nas operações de junção e separação.

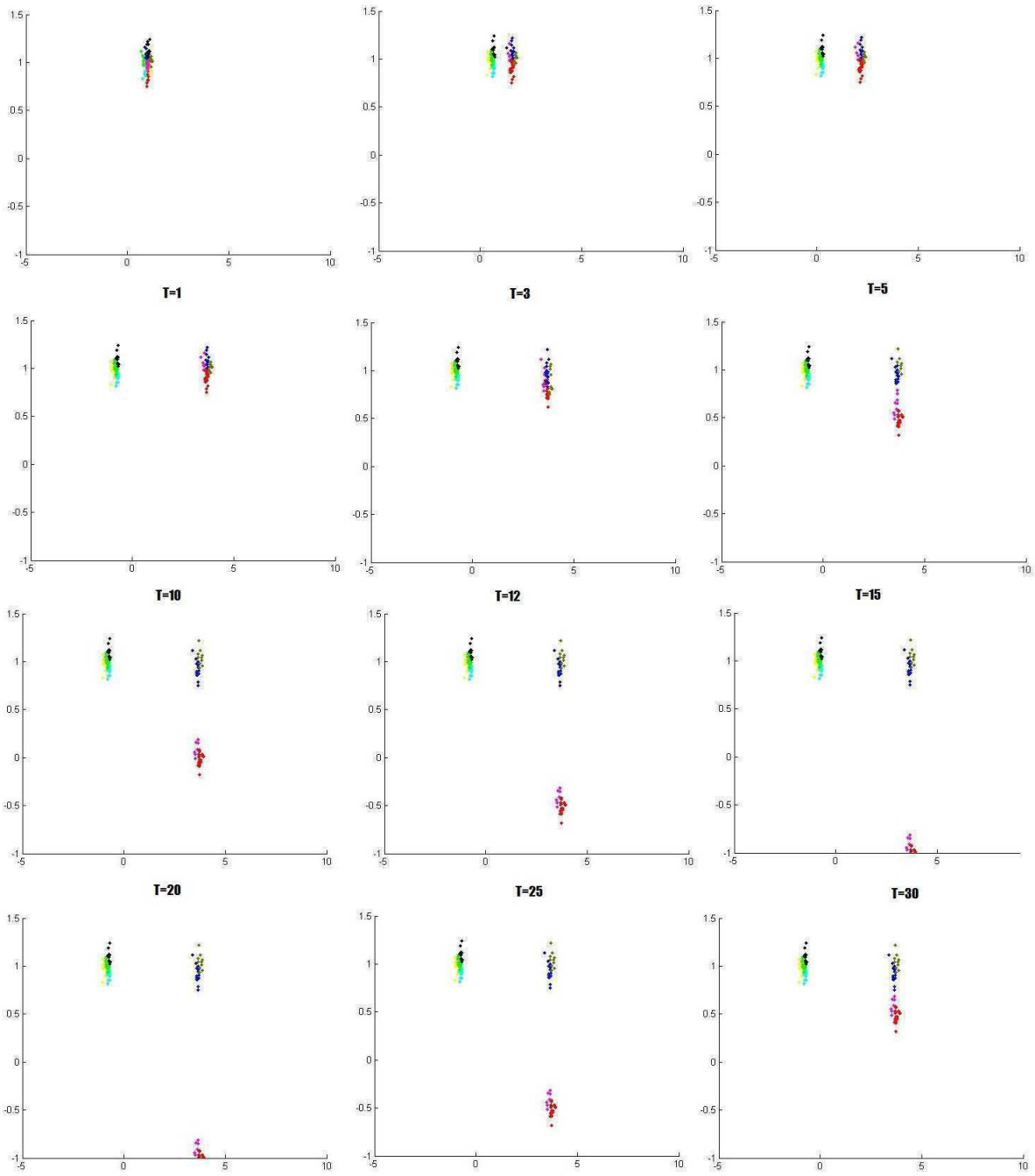


Figura 5.5: Evolução do conjunto de teste ao longo do tempo

5.3 Comportamento das Medidas de Proximidade

Nessa seção detalharemos como cada medida de proximidade (tradicionais e baseadas na teoria da informação) se comportou à medida que os dados do conjunto de teste evoluiram no tempo. Neste trabalho, foram analisadas as medidas de distância euclidiana, distância de Mahalanobis, o potencial de informação e a QRNS.

A metodologia para o experimento consistiu em realizar uma quantização vetorial nos dados a cada instante de tempo por meio do algoritmo k-means com $k = 25$. Essa quantização divide o conjunto de dados em 25 clusters auxiliares, cada um com seus respectivos centros. Esses centros são os pontos representativos utilizados para o cálculo das medidas de proximidade, reduzindo consideravelmente a dimensão dos dados envolvidos. Essa clusterização inicial tem o objetivo apenas de viabilizar a implementação do cálculo das medidas bem como reduzir a dimensão dos dados, porém não influencia na quantidade real de clusters existentes entre os dados.

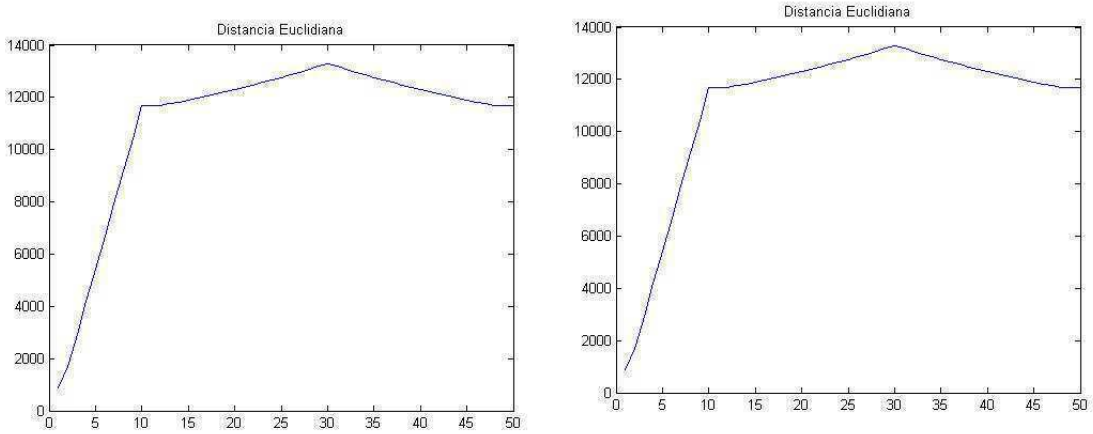
Uma das contribuições desse trabalho é investigar o comportamento dessas medidas ao longo do tempo em processos dinâmicos para então apresentar metodologias para o seu uso na fase de detecção de divisões e junções de clusters. Cada uma das medidas foi calculada a cada instante de tempo, e uma análise foi realizada no seu comportamento mediante a evolução dos dados. À medida que os pontos da nuvem de dados se distanciam ou se aproximam, nota-se uma mudança na curva dessas medidas e é a partir dessas mudanças que detectaremos as operações de junção e separação entre os clusters. O comportamento de cada medida estudada é apresentado nas subseções a seguir.

5.3.1 Distância Euclidiana

O uso da distância euclidiana como critério nas operações de junção e separação entre clusters consiste em monitorar a medida ao longo do tempo até que alguma condição limite se estabeleça, gerando então um alarme que indica que alguma mudança ocorreu na estrutura dos agrupamentos. A distância euclidiana considerada no contexto desse trabalho refere-se à soma das distâncias entre cada par de pontos do conjunto de teste. Esse cálculo gera um custo computacional considerável, tornando-se impraticável à medida que a instância do problema aumenta. Diante disso, nesse trabalho utilizou-se os centros dos clusters auxiliares obtidos na fase de quantização como os pontos representativos para o cálculo da medida. A figura 5.6 apresenta o comportamento da distância euclidiana para a simulação utilizando todo o conjunto de teste (a) e considerando-se apenas os pontos representativos do conjunto (b).

Note que as curvas na figura 5.6 apresentam comportamentos bem similares, o que nos leva a concluir que podemos usar apenas os pontos representativos no cálculo da medida, reduzindo consideravelmente o custo computacional envolvido. Analisando ainda a figura 5.6, percebe-se que a curva apresenta dois vértices nos instantes T=10 e T=30. Esses vértices marcam exatamente os instantes em que a nuvem de pontos começa a se movimentar de forma diferente, indicando uma futura divisão ou junção de clusters, que segundo apresentado na seção 5.2 acontecem nos instantes T=3, T=15 e T=50. Perceba que não observamos nenhum vértice nos primeiros instantes de simulação, o que dificulta a detecção da separação entre clusters no instante T=3. Diante disso, apresentamos na figura 5.7 o comportamento da derivada da distância euclidiana ao longo do tempo.

Note na figura 5.7 que existem 3 grandes quedas na curva com os valores da derivada. São exatamente nesses instantes onde se iniciam as divisões e junções de clusters. Essas quedas nos valores ocorrem nos mesmos instantes onde observamos os vértices da figura 5.6 e além disso, ainda podemos observar o instante onde acontece a primeira divisão



(a) Calculada utilizando todo o conjunto de teste. (b) Calculada utilizando apenas os pontos representativos

Figura 5.6: Comportamento da distância euclidiana do conjunto de teste ao longo do tempo.

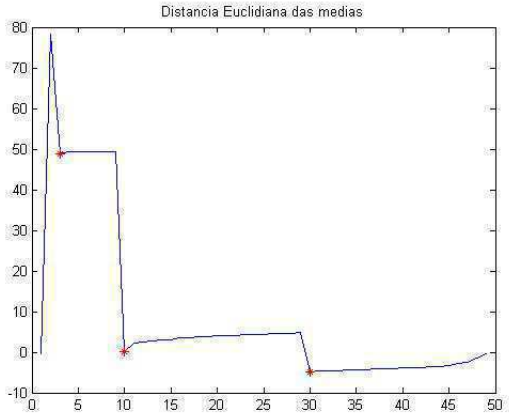


Figura 5.7: Comportamento da derivada da distância euclidiana ao longo do tempo

(T=3) com mais clareza. Diante disso, podemos observar que as separaçoes e as junções de clusters estão associadas às quedas de valores na derivada da distância euclidiana do conjunto. Em outras palavras, se o valor da derivada apresentar uma queda acima de um certo limiar pré-estabelecido, temos uma separação ou junção de clusters. O limiar utilizado no algoritmo desse trabalho foi de 70%.

Como descrito na seção 3.3.1, a distância euclidiana entre dois pontos é dada pela equação 5.2.

$$D_e(x_i, x_j) = \left(\sum_{l=1}^{d} |x_{il} - x_{jl}|^{\frac{1}{2}} \right)^2 \tag{5.2}$$

onde x_i e x_j são objetos de dados d-dimensionais.

Para calcular a soma das distâncias entre cada par de pontos do conjunto de dados deve-se usar a equação 5.3.

$$D_e(X) = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\sum_{l=1}^{d} |x_{il} - x_{jl}|^{\frac{1}{2}} \right)^2 \tag{5.3}$$

onde X é o vetor com os pontos representativos do conjunto de dados e N é a quantidade de pontos em X.

Para que haja um monitoramento do processo dinâmico, a medida deve ser calculada a cada instante de tempo, logo devemos incorporar a variável tempo na equação.

$$D_e(X_t) = \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \left(\sum_{l=1}^{d} |x_{il} - x_{jl}|^{\frac{1}{2}} \right)^2 \tag{5.4}$$

onde X_t é o vetor com os pontos representativos do conjunto de dados no instante t e N_t é a quantidade de pontos em X_t.

Os valores das medidas a cada instante de tempo formam uma série temporal, a qual analisamos seu comportamento ao longo do tempo. A variável D_e representa essa série temporal na equação 5.5.

$$D_e = \begin{pmatrix} D_e(X_1) \\ D_e(X_2) \\ \vdots \\ D_e(X_t) \end{pmatrix} \tag{5.5}$$

Para facilitar a análise da série temporal, fez-se necessário o cálculo da derivada D'_e, onde:

$$D'_e(X_w) \lim_{x \to 0} = \frac{D_e(X_w) - D_e(X_{w-1})}{\Delta x} \tag{5.6}$$

onde $w = \{2 \ldots t\}$

Logo temos uma outra série temporal representada na equação 5.7.

$$D'_e = \begin{pmatrix} - \\ D'_e(X_2) \\ \vdots \\ D'_e(X_t) \end{pmatrix} \tag{5.7}$$

O algoritmo 1, apresenta as regras para identificação de fusões e de separações entre clusters a partir da observação da derivada da distância euclidiana ao longo do tempo.

Algoritmo 1: JUNÇÃO E SEPARAÇÃO POR MEIO DA DERIVADA DA DISTÂNCIA EUCLIDIANA

Entrada: *Limiar*

início
  while *true* **do**
    X_t = conjunto com os pontos representativos no instante atual t
    $D_e(X_t)$ = a soma da distância euclidiana de acordo com a equação 5.4:
    Inclua $D_e(X_t)$ no vetor de distâncias D_e
    Calcule o vetor de derivadas D'_e
    if $D'_e(X_t) < D'_e(X_{t-1}) * Limiar$ **then**
      Dispare Alarme = "Separação ou Junção entre Clusters"
    end
  end
fim

Note que o algoritmo necessita receber como parâmetro de entrada o valor de um limiar que representa o percentual de queda de valores na derivada, o qual caracterizará uma junção ou separação entre os clusters.

5.3.2 Distância Mahalanobis

Semelhante ao algoritmo com a distância euclidiana, para a distância de Mahalanobis também considera-se a soma das distâncias entre cada par de pontos do conjunto de teste e com o intuito de diminuir o custo computacional, utilizou-se os centros dos clusters auxiliares obtidos na fase de quantização como os pontos representativos para o cálculo da medida.

Conforme definido na seção 3.3.1, a distância de Mahalanobis entre dois pontos é calculada segundo a equação 5.8:

$$D_m(x_i, x_j) = (x_i - x_j)^T S^{-1} (x_i - x_j) \tag{5.8}$$

onde S é a matriz de covariância dentro da classe definida.

Para o cálculo da soma das distâncias entre todos os pares de pontos do conjunto, deve-se utilizar a equação 5.9.

$$D_m(X) = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(x_i - x_j\right)^T S^{-1} \left(x_i - x_j\right) \tag{5.9}$$

onde X é o vetor com os pontos representativos do conjunto de dados e N é a quantidade de pontos em X. Incorporando a variável tempo na equação, temos:

$$D_m(X_t) = \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \left(x_i - x_j\right)^T S^{-1} \left(x_i - x_j\right) \tag{5.10}$$

onde X_t é o vetor com os pontos representativos do conjunto de dados no instante t e N_t é a quantidade de pontos em X_t.

A série temporal com os valores das distâncias em cada instante de tempo é representada por 5.11:

$$D_m = \begin{pmatrix} D_m(X_1) \\ D_m(X_2) \\ \vdots \\ D_m(X_t) \end{pmatrix} \tag{5.11}$$

A derivada D'_m, é dada por:

$$D'_m(X_w) \lim_{x \to 0} = \frac{D_m(X_w) - D_m(X_{w-1})}{\Delta x} \tag{5.12}$$

onde $w = \{2 \ldots t\}$

A série temporal com os valores da derivada de D_m é representada na equação 5.13.

$$D'_m = \begin{pmatrix} - \\ D'_m(X_2) \\ \vdots \\ D'_m(X_t) \end{pmatrix} \tag{5.13}$$

A figura 5.8 apresenta o comportamento da distância de Mahalanobis ao longo do tempo.

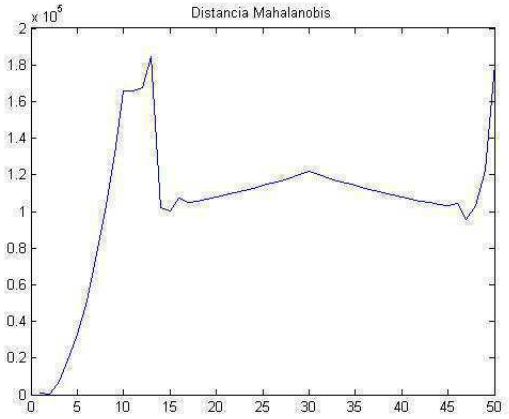


Figura 5.8: Comportamento da distância de Mahalanobis do conjunto de teste 1 ao longo do tempo

Note que temos três curvas crescentes ao longo do tempo. Elas se iniciam nos instantes T=3, T=15 e T=48, as quais marcam mudanças significativas no comportamento dos pontos e consequentemente futuras divisões e junções entre clusters. Uma divisão ou junção entre clusters acontece no início das curvas crescentes.

O algoritmo 2, apresenta as regras para identificação de junções e de separações entre clusters a partir da observação da distância de mahalanobis ao longo do tempo. A variável *RegiaoEstavel* tem a função de verificar se estamos diante do início de uma nova curva crescente.

Algoritmo 2: JUNÇÃO E SEPARAÇÃO POR MEIO DA DISTÂNCIA DE MAHALANOBIS

```
   Entrada: Limiar
1  início
2      while true do
3          X_t = conjunto com os pontos representativos no instante atual t
4          D_m(X_t) = a soma da distância de mahalanobis de acordo com a equação
             5.10
5          Inclua D_m(X_t) no vetor de distância D_m
6          if RegiaoEstavel = False then
7              if (D_m(X_t) > D_m(X_{t-1}) then
8                  Dispare Alarme = "Junção ou Separação entre Clusters"
9              end
10         end
11     end
12 fim
```

A figura 5.9 apresenta o comportamento da derivada da distância de Mahalanobis ao longo do tempo.

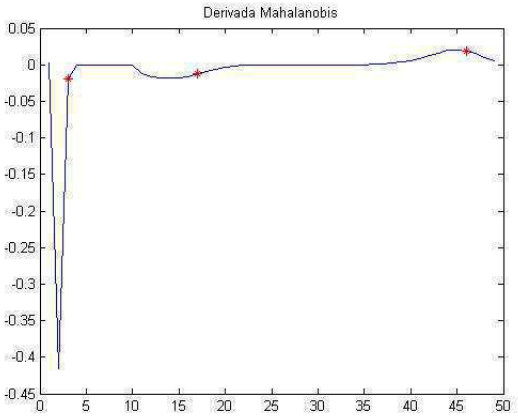


Figura 5.9: Comportamento da derivada da distância de Mahalanobis ao longo do tempo

Ao analisarmos os valores da derivada, podemos detectar que as fusões e divisões de clusters ocorrem no início de uma série de valores positivos. Em outras palavras, essas operações acontecem quando há uma mudança de valores negativos para positivos na derivada.

O algoritmo 3, apresenta as regras para identificação de junções e de separações entre clusters a partir da observação da derivada da distância de Mahalanobis ao longo do tempo.

Algoritmo 3: JUNÇÃO E SEPARAÇÃO POR MEIO DA DERIVADA DA DISTÂNCIA DE MAHALANOBIS

```
1  início
2      while true do
3          X_m = conjunto com os pontos representativos no instante atual t
4          D_m(X_t) = a soma da distância de mahalanobis de acordo com a equação
             5.10
5          Inclua D_m(X_t) no vetor de distância D_m
6          Calcule o vetor de derivadas D'_m
7          if D'_m(X_t) >= 0 e D'_m(X_{t-1}) < 0 then
8              Dispare Alarme = "Junção ou Separação entre Clusters"
9          end
10     end
11 fim
```

5.3.3 Potencial de Informação (IP)

Para a análise do IP foram avaliadas duas situações: utilizando todos os pontos para o cálculo e utilizando apenas os pontos representativos do conjunto para o cálculo. A figura 5.10 apresenta o comportamento do IP ao longo do tempo para essas duas situações.

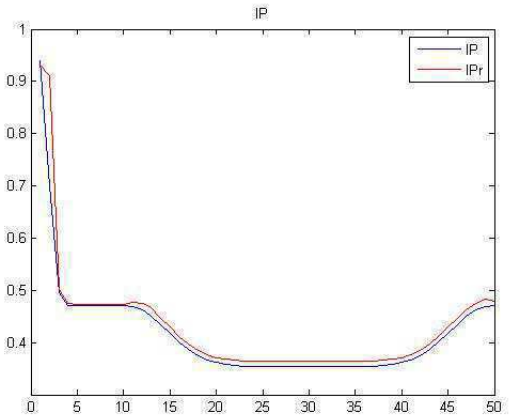


Figura 5.10: Comportamento do potencial de informação do conjunto de teste 1 ao longo do tempo

Note que as duas curvas do comportamento do IP na figura 5.10 são bem parecidas, o

que nos deixa concluir que podemos calcular o IP utilizando apenas os pontos representativos, reduzindo assim o custo computacinal significativamente.

Analisando a curva do comportamento do IP, verificamos que a cada divisão de clusters, a curva se torna descendente e a cada junção de clusters, a curva se torna ascendente. Em outras palavras, ao detectarmos uma descida considerável no valor do IP, significa que estamos próximos a uma divisão de clusters e quando nos deparamos com uma subida no valor do IP estamos diante de uma junção entre clusters. Isso pode observado entre os instantes T=1 e T=5, T=10 e T=15 e T=40 e T=50. A maior dificuldade ao se trabalhar com o IP, é que esse método é dependente dos parâmetros que definem o limiar de subida e de descida. Esse limiar pode variar de acordo com os dados, o que não é uma característica desejavél nesses algoritmos. O limiar utilizado nesse trabalho foi de 15% para as quedas e de 6% para as subidas.

Conforme apresentado na seção 2.1, o potencial de informação do conjunto de dados foi calculado de acordo com a equação 5.14.

$$D_{IP}(X) = \left(\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_j - x_i) \right) \tag{5.14}$$

onde X é o vetor com os pontos representativos do conjunto e N é a quantidade de pontos em X. Incoporando a variável tempo, temos:

$$D_{IP}(X_t) = \left(\frac{1}{N_t^2} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} G_{\sigma\sqrt{2}}(x_j - x_i) \right) \tag{5.15}$$

onde X_t é o vetor com os pontos representativos do conjunto de dados no instante t e N_t é a quantidade de pontos em X_t.

A série temporal com os valores das distâncias em cada instante de tempo é representada por 5.16:

$$D_{IP} = \begin{pmatrix} D_{IP}(X_1) \\ D_{IP}(X_2) \\ \vdots \\ D_{IP}(X_t) \end{pmatrix} \tag{5.16}$$

A derivada D'_{IP}, é dada por:

$$D'_{IP}(X_w) \lim_{x \to 0} = \frac{D_{IP}(X_w) - D_{IP}(X_{w-1})}{\Delta x} \tag{5.17}$$

onde $w = \{2 \ldots t\}$

A série temporal com os valores da derivada de D_{IP} é representada na equação 5.18.

$$D'_{IP} = \begin{pmatrix} - \\ D'_{IP}(X_2) \\ \vdots \\ D'_{IP}(X_t) \end{pmatrix} \tag{5.18}$$

O algoritmo 4, apresenta as regras para identificação de junções e de separações entre clusters a partir da observação do IP ao longo do tempo.

Algoritmo 4: JUNÇÃO E SEPARAÇÃO POR MEIO DO IP

Entrada: $LimiarDescida, LimiarSubida$

início
  while *true* **do**
    X_t = conjunto com os pontos representativos no instante atual t
    $D_{IP}(X_t)$ = Calcule o IP no instante t de acordo com a equação 5.15
    Inclua $D_{IP}(X_t)$ no vetor de distâncias D_{IP}
    if $D_{IP}(X_t) < D_{IP}(X_{t-1}) * LimiarDescida$ **then**
      Dispare Alarme = "Separação de Cluster"
    end
    if $D_{IP}(X_t) > D_{IP}(X_{t-1}) * LimiarSubida$ **then**
      Dispare Alarme = "Junção de Cluster "
    end
  end
fim

A figura 5.11 apresenta o comportamento da derivada do potencial de informação ao longo do tempo. Para a detecção de fusões e divisões por meio da derivada do IP, também devemos definir um limiar com intervalo próximo a zero, nesse trabalho utilizou-se o intervalo $-10^{-2} < X < 10^{-2}$. A cada nova entrada dentro do intervalo limiar detecta-se uma separação ou junção entre clusters. Uma vez que a curva ainda apresente valores dentro desse intervalo, considera-se que a mesma se encontra em uma região estável. Um novo ponto de detecção de separação ou junção só será observado após uma saída dessa região estável e nova entrada. O próximo valor que estiver dentro do limiar, será um novo instante de divisão ou junção. Na figura 5.11, podemos observar que os instantes T=4, T=16 e T=48 satifazem a condição e marcam essas operações de divisão ou junção.

O algoritmo 5, apresenta as regras para identificação de fusões e de separações entre clusters a partir da observação da derivada do IP ao longo do tempo.

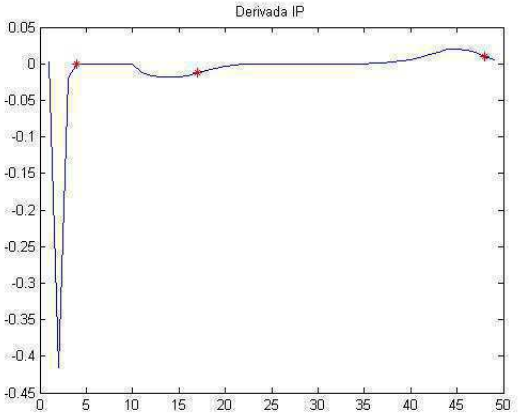


Figura 5.11: Comportamento da derivada do potencial de informação ao longo do tempo

Algoritmo 5: JUNÇÃO E SEPARAÇÃO POR MEIO DA DERIVADA DO IP

Entrada: $LimiarInferior, LimiarSuperior$

início
  while *true* **do**
    X_t = conjunto com os pontos representativos no instante atual t
    $D_{IP}(X_t)$ = IP no instante t de acordo com a equação 5.15
    Inclua $D_{IP}(X_t)$ no vetor de distâncias D_{IP}
    Calcule o vetor de derivadas D'_{IP}
    if $RegiaoEstavel = false$ **then**
      if $D'_{IP}(X_t) > LimiarInferior$ *e* $D'_{IP}(X_t) < LimiarSuperior$ **then**
        Dispare Alarme = "Junção ou Separação entre Clusters"
      end
    end
  end
fim

5.3.4 QRNS

Conforme apresentado na seção 3.3.2, a QRNS entre duas gaussianas é calculada de acordo com a equação 5.19.

$$D_{QRNS}(x_1,x_2) = \log\left(\frac{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(P_1N(x_1,\mu_1,\Sigma_1)+P_2N(x_2,\mu_2,\Sigma_2))^2dx}{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}N(x,\mu_q,\Sigma_q)^2dx}\right) \tag{5.19}$$

onde

$$\mu_q = P_1\mu_1 + P_2\mu_2 \tag{5.20}$$

$$\Sigma_q = P_1\Sigma_1 + P_2\Sigma_2 + P_1P_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)^t \tag{5.21}$$

No contexto desse trabalho, para a detecção de junções e separaçoes entre clusters, avalia-se a soma das QRNS's entre cada par de gaussianas no conjunto de teste. Nesse caso, cada clusters auxiliar encontrado na fase de quantização representa uma gaussiana. Para o calculo da soma de todas das medidas entre os pares de clusters, deve-se usar a equação 5.22.

$$D_{QRNS}(X) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\log\left(\frac{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(P_iN(x_i,\mu_i,\Sigma_i)+P_jN(x_j,\mu_j,\Sigma_j))^2dx}{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}N(x,\mu_q,\Sigma_q)^2dx}\right)\right) \tag{5.22}$$

onde

$$\mu_q = P_i\mu_i + P_j\mu_j$$

$$\Sigma_q = P_i\Sigma_i + P_j\Sigma_j + P_iP_j(\mu_i - \mu_j)(\mu_i - \mu_j)^t$$

e n é a quantidade clusters no conjunto.

Incoporando a variável tempo, temos:

$$D_{QRNS}(X_t) = \sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\left(\log\left(\frac{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(P_iN(x_i,\mu_i,\Sigma_i)+P_jN(x_j,\mu_j,\Sigma_j))^2dx}{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}N(x,\mu_q,\Sigma_q)^2dx}\right)\right) \tag{5.23}$$

e n_t é a quantidade clusters no conjunto no instante t.

A série temporal com os valores das distâncias em cada instante de tempo é representada por 5.24:

$$D_{QRNS} = \begin{pmatrix} D_{QRNS}(X_1) \\ D_{QRNS}(X_2) \\ \vdots \\ D_{QRNS}(X_t) \end{pmatrix} \tag{5.24}$$

A derivada D'_{QRNS}, é dada por:

$$D'_{QRNS}(X_w)\lim_{x\to 0} = \frac{D_{QRNS}(X_w) - D_{QRNS}(X_{w-1})}{\Delta x} \tag{5.25}$$

onde $w = \{2\ldots t\}$

A série temporal com os valores da derivada de D_{IP} é representada na equação 5.26.

$$D'_{QRNS} = \begin{pmatrix} - \\ D'_{QRNS}(X_2) \\ \vdots \\ D'_{QRNS}(X_t) \end{pmatrix} \tag{5.26}$$

A figura 5.12 apresenta o comportamento da medida ao longo do tempo.

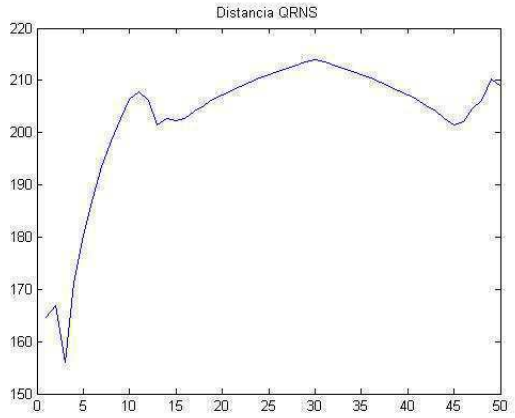


Figura 5.12: Comportamento do QRNS do conjunto de teste 1 ao longo do tempo

Na análise da curva do QRNS, percebe-se que as divisões ou junções de clusters acontecem exatamente no início de curvas ascendentes, ou seja, sempre que inicia-se um novo crescimento nos valores da medida, estamos diante de separações ou junções de clusters. Na figura 5.12, podemos observar que os instantes T=3, T=15 e T=47 satisfazem a condição.

O algoritmo 6, apresenta as regras para identificação de fusões e de separações entre clusters a partir da observação do QRNS ao longo do tempo.

Algoritmo 6: JUNÇÃO E SEPARAÇÃO POR MEIO DO QRNS

```
1  início
2      while true do
3          X_t = conjunto com os clusters auxiliares no instante atual t
4          D_QRNS(X_t) = QRNS do conjunto no instante t de acordo com a equação
             5.23:
5          Inclua D_QRNS(X_t) no vetor D_QRNS
6          if RegiaoEstavel = False then
7              if D_QRNS(X_t) > D_QRNS(X_{t-1}) then
8                  Dispare Alarme = "Junção ou Separação entre Clusters"
9              end
10         end
11     end
12 fim
```

A figura 5.13 apresenta o comportamento da derivada do QRNS ao longo do tempo.

Assim como na distância de Mahalanobis, podemos detectar as junções e divisões pela observação do início de valores positivos na derivada do QRNS. Essas operações acontecem quando há uma mudança de valores negativos para positivos na derivada. Os

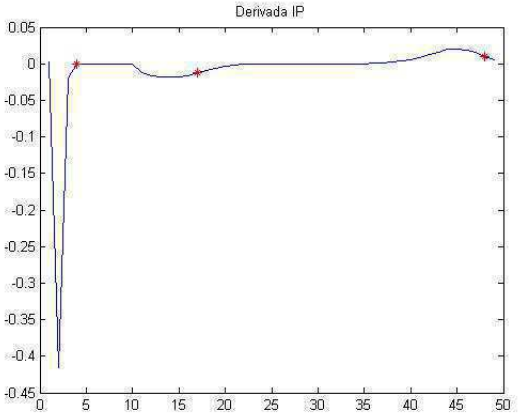


Figura 5.13: Comportamento da derivada do potencial de informação ao longo do tempo

instantes detectados pela observação da derivada na figura 5.13 são os mesmos da figura 5.12: T=3, T=15 e T=47.

O algoritmo 7, apresenta as regras para identificação de junções e de separações entre clusters a partir da observação da derivada do QRNS ao longo do tempo.

Algoritmo 7: JUNÇÃO E SEPARAÇÃO POR MEIO DA DERIVADA DO QRNS

```
1  início
2      while true do
3          X_t = conjunto com os clusters auxiliares no instante atual t
4          D_QRNS(X_t) = QRNS do conjunto no instante t de acordo com a equação
             5.23:
5          Inclua D_QRNS(X_t) no vetor D_QRNS
6          Calcule o vetor de derivadas D'_QRNS
7          if D'_QRNS(X_t) >= 0 e D'_QRNS(X_{t-1}) < 0 then
8              Dispare Alarme = "Junção ou Separação entre Clusters"
9          end
10     end
11 fim
```

Em resumo, as regras utilizadas para cada algoritmo são:

- Distância Euclidiana (Derivada): Se o valor da derivada apresentar uma queda acima de um certo limiar pré- estabelecido, significa que temos uma separação ou junção de cluster. O limiar utilizado no algoritmo foi de 70%, ou seja, se o valor da derivada cair acima de 70% em relação ao valor no instante anterior, temos uma separação ou junção entre clusters.

- Distância de Mahalanobis: As separações ou junções entre clusters são detectadas a cada nova subida de valores, ou seja, no início de movimento crescente na curva.

- Distância de Mahalanobis (Derivada): As separações ou junções são detectadas o primeiro valor positivo encontrado após uma sequencia de valores negativos na curva.

- IP: Se o valor do IP apresentar uma queda ou subida acima do limiar pré-estabelecido teremos uma separação(queda) ou junção(subida) entre clusters. O limiar utilizado no algoritmo foi de 15% para as quedas e 6% para as subidas.

- IP (Derivada): Temos uma nova separação ou junção sempre que a curva apresentar um valor dentro de um intervalo definido pelo limiar. Uma vez que a curva ainda apresente valores dentro desse intervalo, considera-se que a mesma se encontra em uma região estável. Um novo ponto de detecção de separação ou junção só será observado após a saída dessa região estável e a ocorrência de novo valor dentro do intervalo. O intervalo utilizado nesse algoritmo foi $-10^{-2} < X < 10^{2}$.

- QRNS: Semelhante ao algoritmo da distância de Mahalaobis, as separações ou junções entre clusters são detectadas a cada nova subida de valores, ou seja, no início de movimento crescente na curva.

- QRNS (Derivada): As separações ou junções são detectadas o primeiro valor positivo encontrado após uma sequencia de valores negativos na curva.

5.4 Experimentos e Resultados

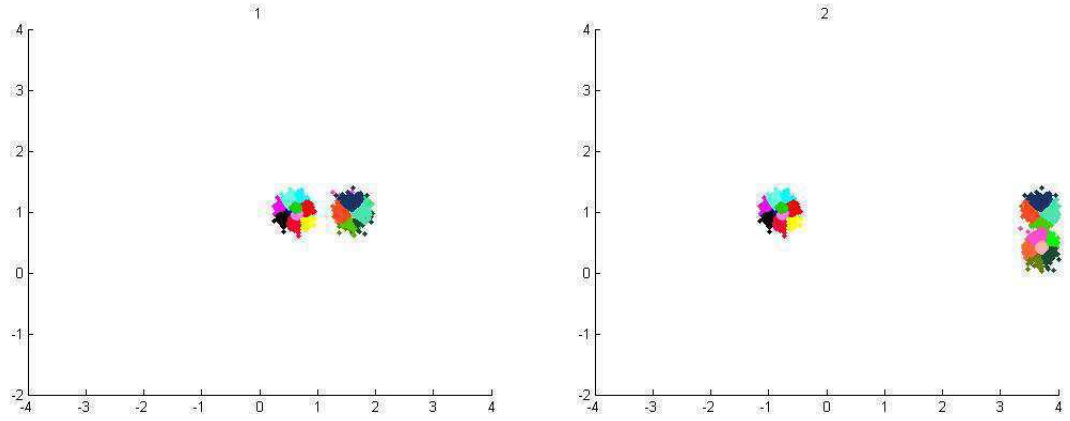
Essa seção apresenta os resultados obtidos com experimentos envolvendo o uso de medidas de informação no contexto de clusters dinâmicos. Neste trabalho, o foco do estudo foi nas operações de junção e divisão de clusters, as quais são utilizadas para a reestruturação dos clusters após a inclusão de novos elementos na nuvem de dados. Os experimentos dessa seção consistiram na comparação do desempenho de medidas de similaridade baseadas na informação e medidas tradicionais para detecção de junções e divisões de clusters. Foram realizados testes com dois conjuntos de treinamento, o primeiro gerado sob distribuições gaussianas e o segundo sob o formato de toroide. Os resultados serão apresentados nas subseções a seguir.

5.4.1 Conjuntos Gaussianos

Nesses primeiros experimentos, foram realizados testes envolvendo conjuntos que foram gerados sob distribuições gaussianas, com médias e variâncias diferentes. À medida que o tempo passa, esses parâmetros são alterados de modo que os clusters formados sejam modificados em forma e quantidade.

Experimento 1: 2 Separações entre clusters

Para o primeiro experimento, os dados foram alterados ao longo do tempo, de modo que ocorressem duas separações entre os clusters, uma no instante T=3 e outra no instante T=16, conforme apresentado na figura 5.14.



(a) Separação no instante T=3 (b) Separação no instante T=16

Figura 5.14: Clusters Dinâmicos - Experimento 1: 2 Separações

Essa simulação foi realizada com conjuntos de 1000, 10.000 e 100.000 pontos com 50 iterações para cada execução. Os resultados são apresentados nas figuras 5.15, 5.16 e 5.17.

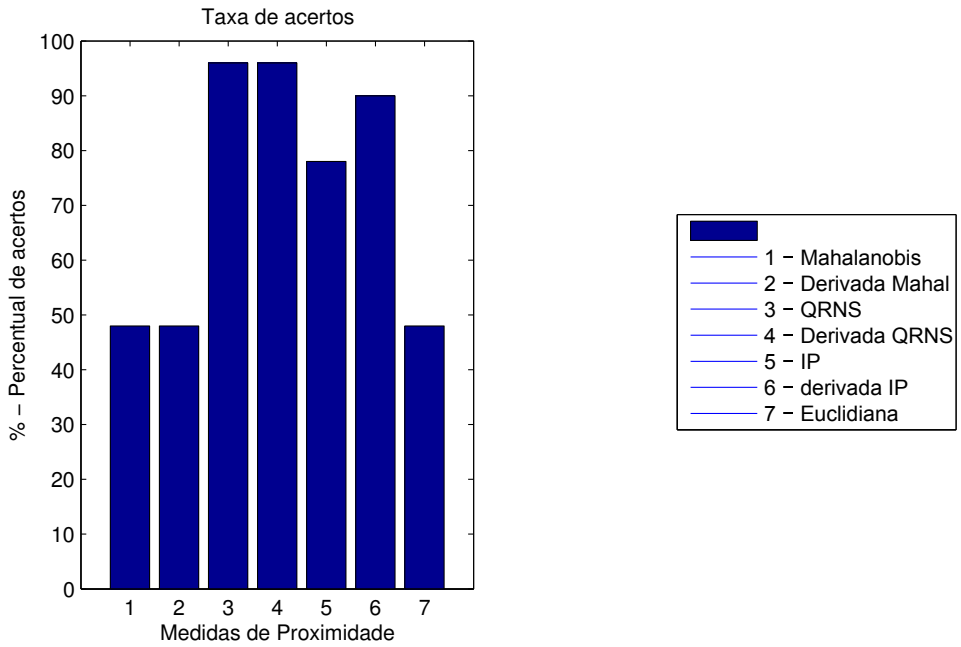


Figura 5.15: Taxa de Acertos - Experimento 1: Simulação com 1.000 pontos

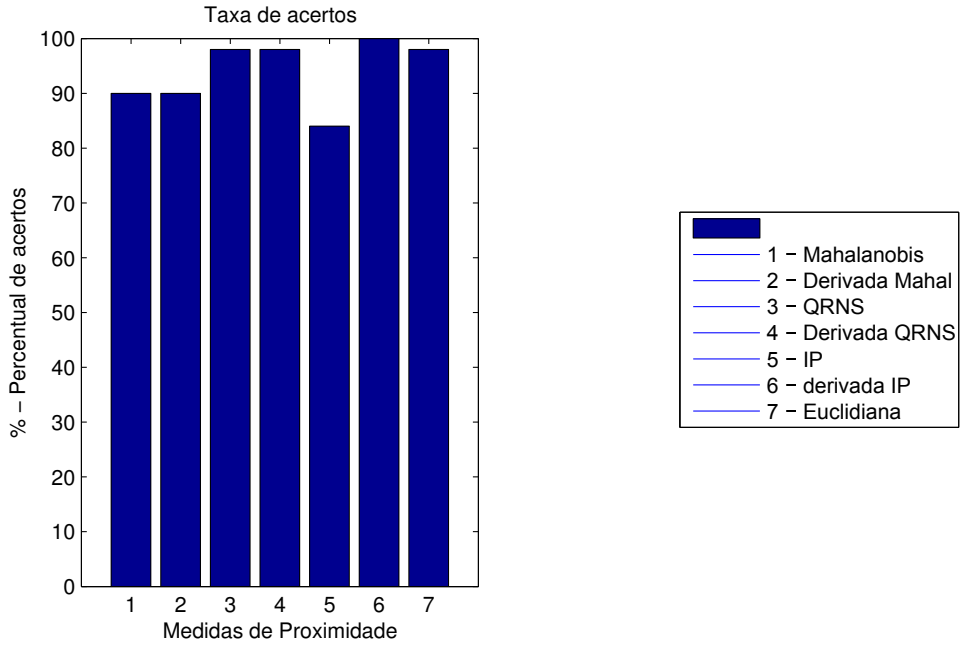


Figura 5.16: Taxa de Acertos - Experimento 1: Simulação com 10.000 pontos

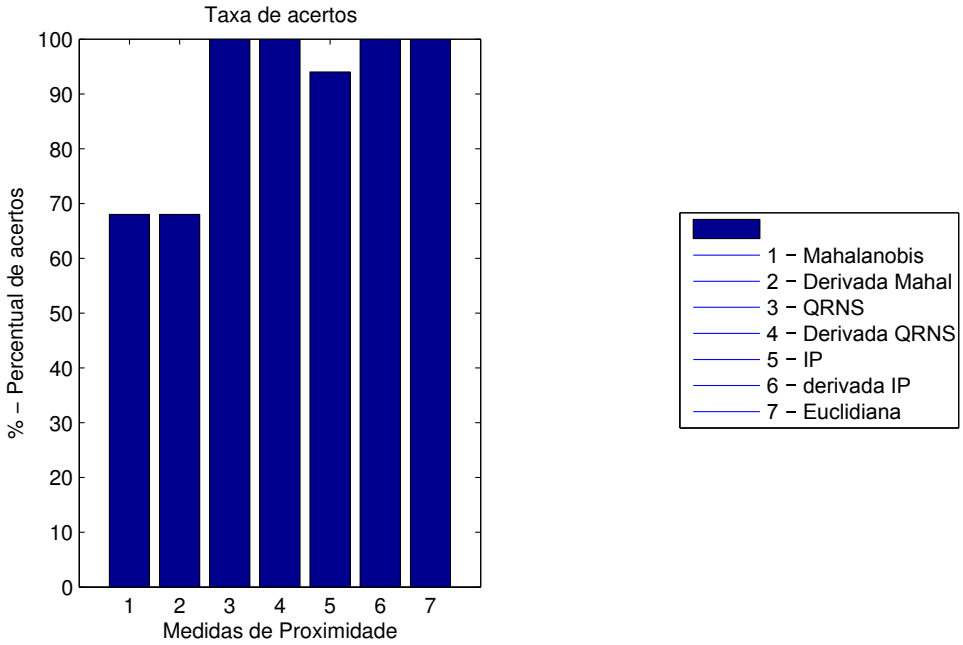


Figura 5.17: Taxa de Acertos - Experimento 1: Simulação com 100.000 pontos

Note que as medidas que obtiveram maior número de acertos foram QRNS e sua derivada, IP e sua derivada e a distância euclidiana. Porém, de uma maneira geral, todos os algoritmos obtiveram uma boa taxa de acerto, à medida que o número de pontos no conjunto de teste aumenta. O número de acertos refere-se à quantidade de vezes que o algoritmo que utiliza a medida em questão acerta a quantidade de divisões existentes na simulação.

As figuras 5.18,5.19 e 5.20 apresentam os instantes médios em que as separações aconteceram em cada algoritmo. A análise dessa figura é muito importante, pois, a partir dela, podemos avaliar a qualidade das medidas no que se respeito à precisão nos instantes de detecção.

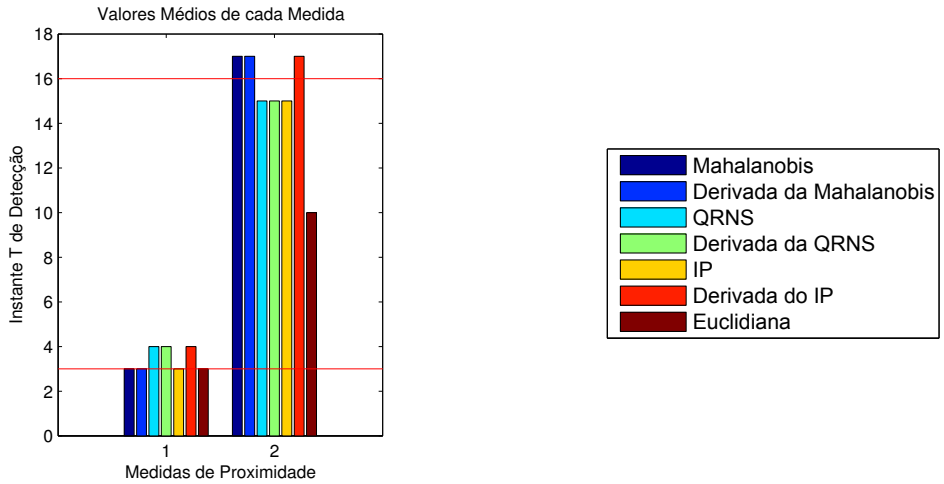


Figura 5.18: Instante Médio das Separações - Experimento 1: Simulação com 1.000 pontos

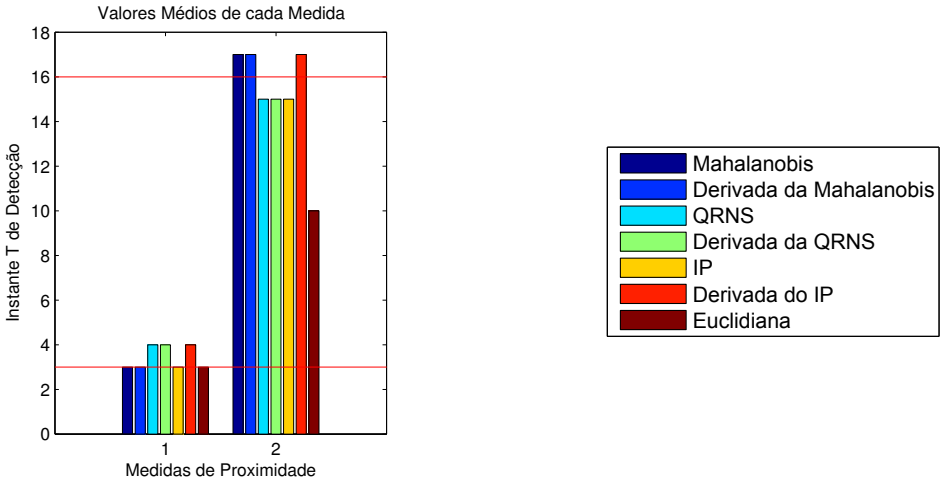


Figura 5.19: Instante Médio das Separações - Experimento 1: Simulação com 10.000 pontos

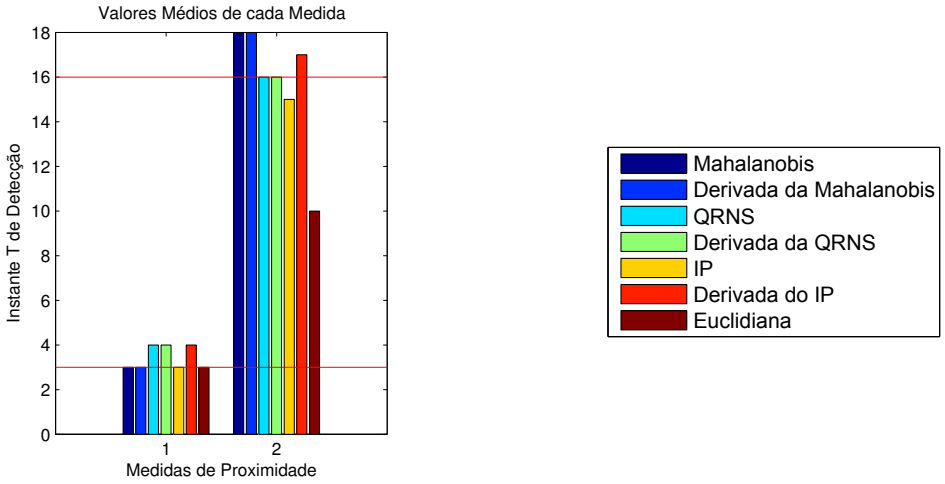
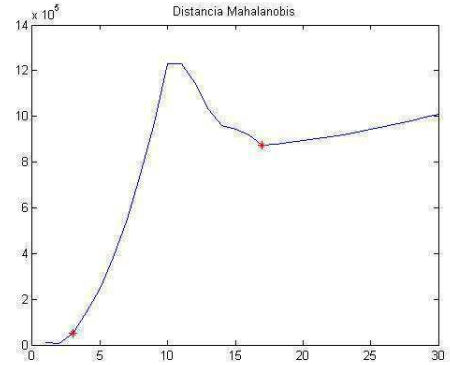


Figura 5.20: Instante Médio das Separações - Experimento 1: Simulação com 100.000 pontos

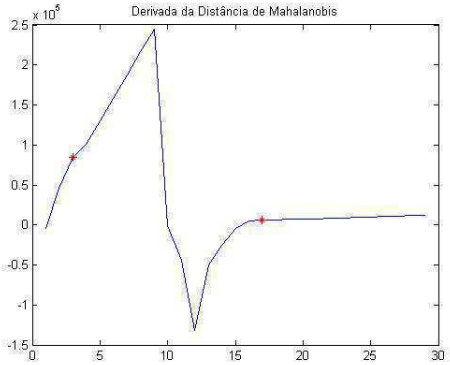
Note que independente da quantidade de pontos no conjunto de teste, o algoritmo baseado na distância euclidiana sempre detecta a segunda separação muito antecipadamente, por volta do instante T=10, quando deveria acontecer por volta do instante T=16. Isso acontece porque o algoritmo da distância euclidiana se baseia em quedas acentuadas na curva da derivada para detectar as junções e separações de clusters. O que acontece é que esses instantes associados às quedas de valores marcam o início da movimentação dos pontos e na maioria das vezes, os clusters ainda não se dividiram por completo. Diante disso, o algoritmo que se baseia na distância euclidiana para detecçoes de separações de clusters não se mostrou adequado para essa simulação.

Nas figuras 5.21 são apresentadas as curvas de cada medida analisada ao longo do tempo, bem como os pontos de separações de clusters detectados por seus algoritmos.

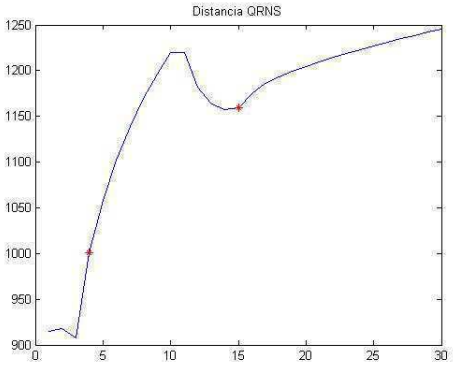
As curvas da figura 5.21 apresentam claramente o resultado das simulações dos algoritmos para cada medida de proximidade, os quais foram implementados seguindo as regras citadas na seção 5.3. Na curva da figura 5.21 (g) podemos confirmar a antecipação na detecção da separação entre clusters utilizando-se a distância euclidiana.



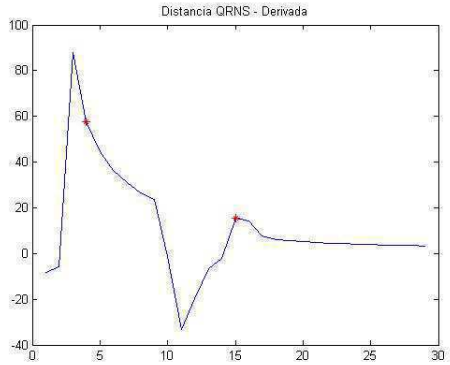
(a) Distância de Mahalanobis



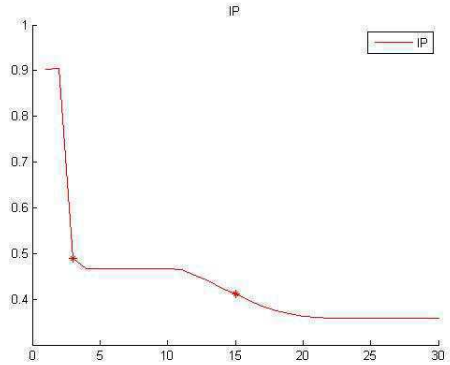
(b) Derivada da Distância de Mahalanobis



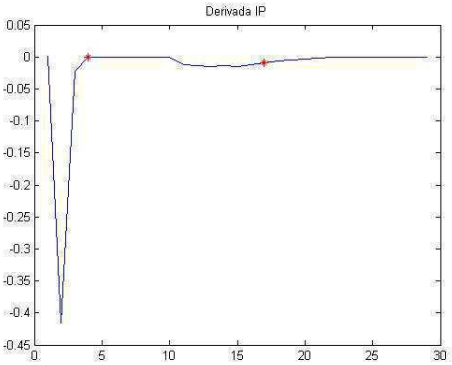
(c) QRNS



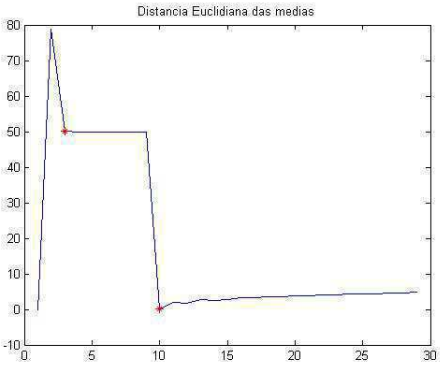
(d) Derivada da QRNS



(e) IP



(f) Derivada do IP

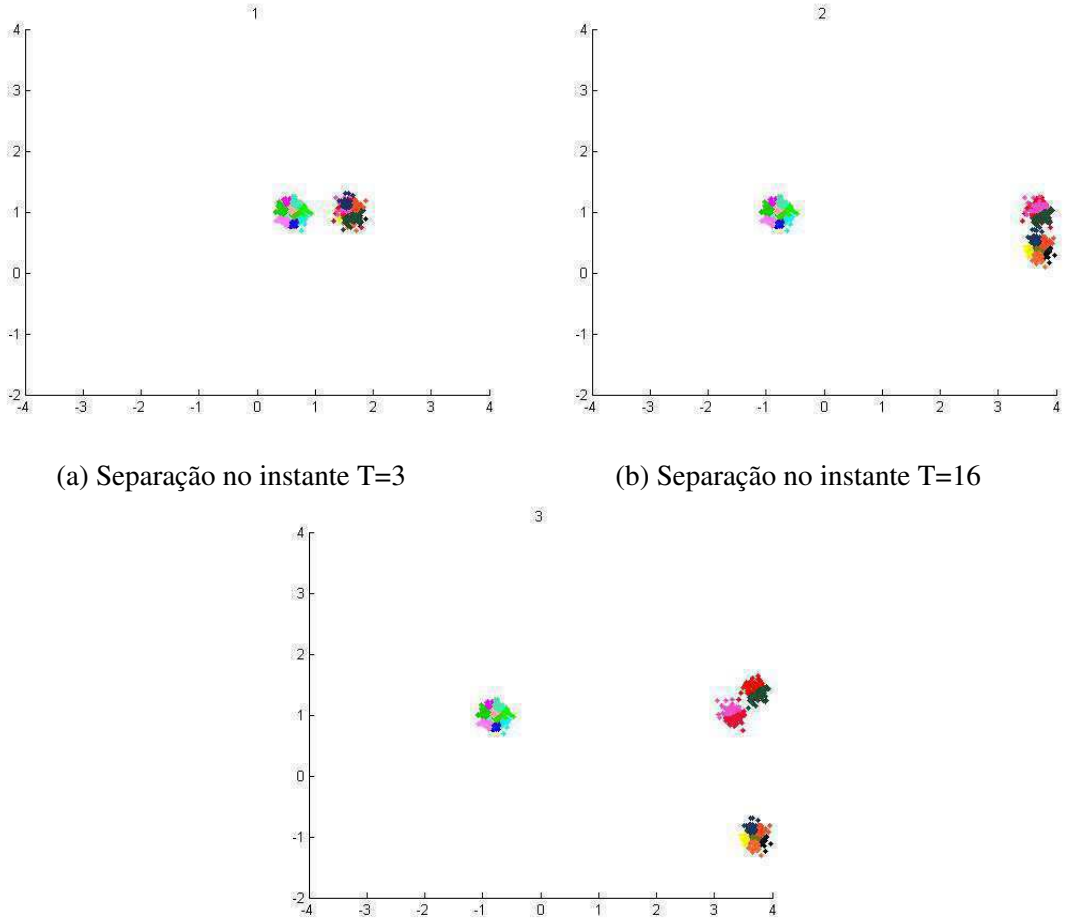


(g) Distancia Euclidiana

Figura 5.21: Comportamento das medidas de similaridade ao longo do tempo. Experimento 1: 2 Separações

Experimento 2: 3 Separações entre clusters

O segundo experimento, consistiu em uma extensão do primeiro. Neste caso, os dados são alterados de modo com que ocorram 3 separações entre clusters: nos instantes T=3, T=16 e T=34. A figura 5.22 apresenta os instantes de separações entre clusters na nuvem de dados.



(a) Separação no instante T=3

(b) Separação no instante T=16

(c) Separação no instante T=34

Figura 5.22: Clusters Dinâmicos - Experimento 2: 3 Separações

Foram executadas 50 iterações para cada simulação e os resultados das simulações para conjuntos com 1000, 10.000 e 100.000 pontos são apresentados nas figuras 5.23, 5.24 e 5.25. Nessas figuras podemos observar claramente que os algoritmos baseados na QRNS e sua derivada e no IP foram os que obtiveram os melhores resultados. Os motivos para a queda de desempenho das demais medidas podem ser esclarecidas pela observação da figura 5.26, onde as curvas com o comportamento de cada medida ao longo do tempo são apresentadas.

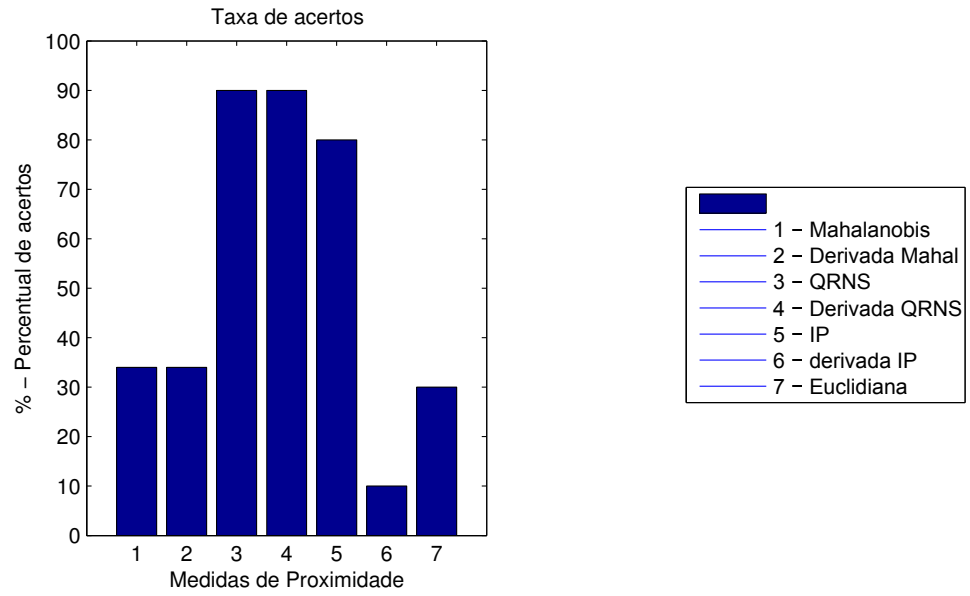


Figura 5.23: Taxa de Acertos - Experimento 2: Simulação com 1.000 pontos

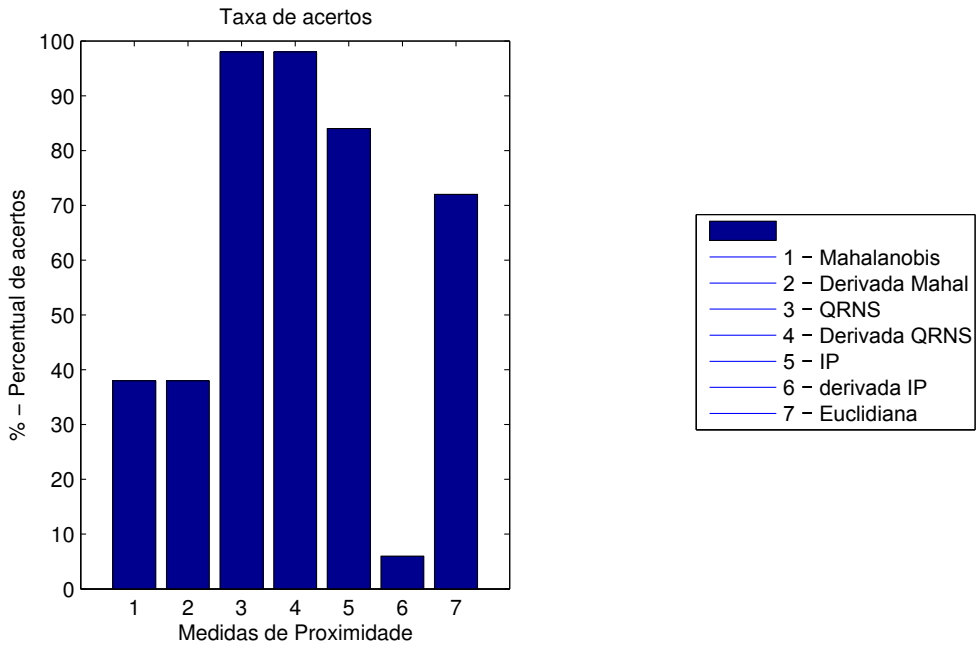


Figura 5.24: Taxa de Acertos - Experimento 2: Simulação com 10.000 pontos

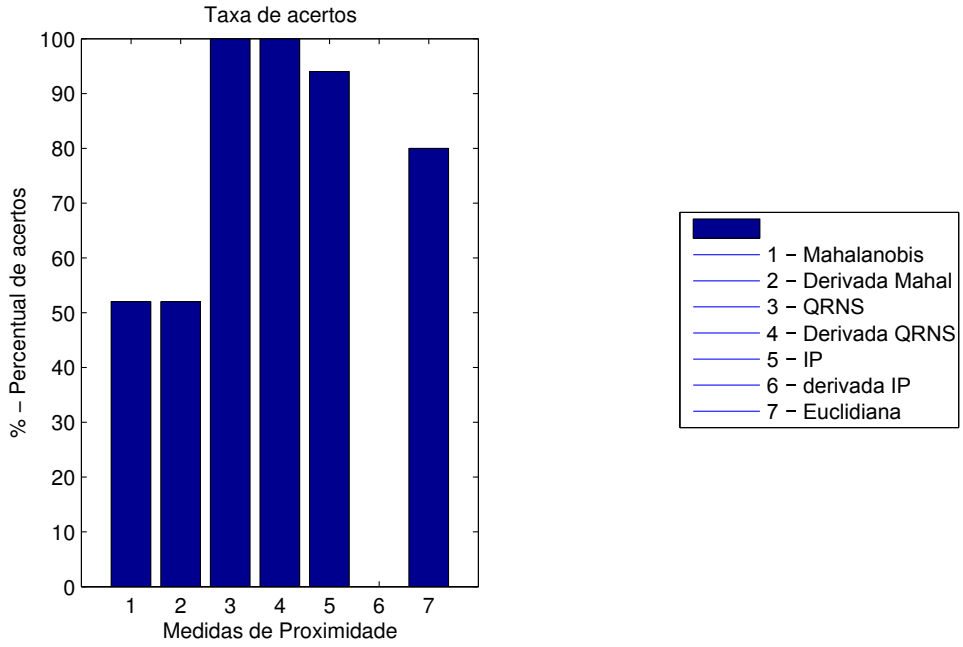
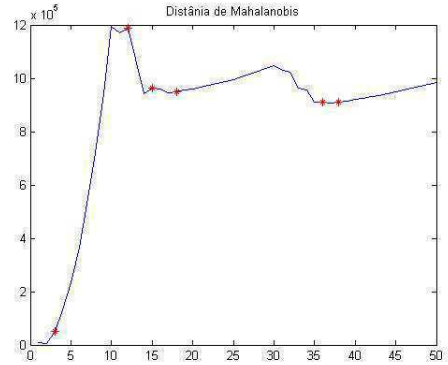


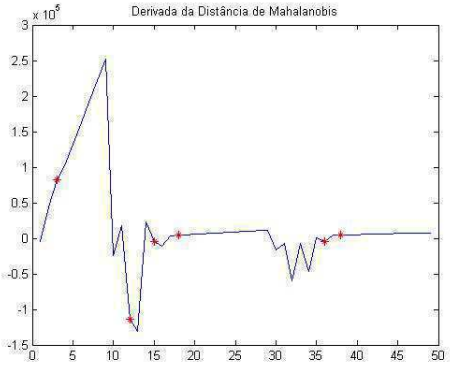
Figura 5.25: Taxa de Acertos - Experimento 2: Simulação com 100.000 pontos

Note na figura 5.26 que os algoritmos baseados na distância euclidiana e mahalanobis detectam pontos extras erroneamente na curva, principalmente no fim da execução. Já o algoritmo baseado na derivada do IP tem dificuldades em encontrar o instante da última separação. Essa dificuldade no algoritmo da derivada do IP, pode ser esclarecida ao observarmos na figura 5.26 em que a variação de valores em sua derivada, relativa à terceira separação, é bem discreta. A figura 5.27 apresenta os instantes médios em que separações aconteceram.

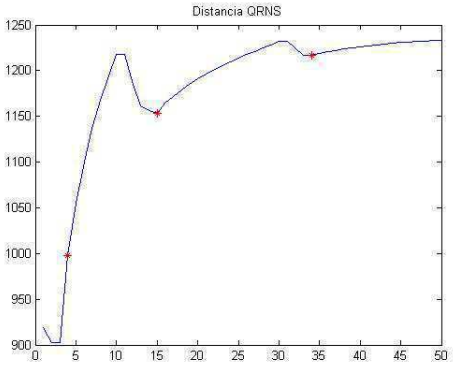
Do mesmo modo do experimento 1, nota-se que o algoritmo da distância euclidiana detecta a segunda separação muito antecipadamente, independentemente da quantidade de pontos no conjunto de teste. E para as simulações com 100.000 pontos, confirma-se o baixo desempenho da derivada do IP. Vale lembrar que esses instantes médios são calculados levando-se em consideração apenas as execuções que acertaram o número de separações de clusters existentes no exemplo.



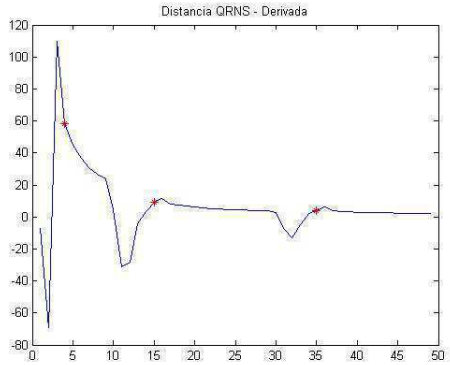
(a) Distância de Mahalanobis



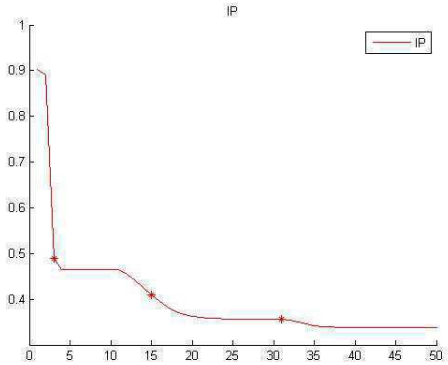
(b) Derivada da Distância de Mahalanobis



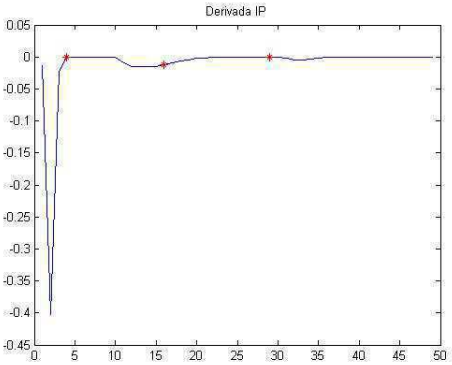
(c) QRNS



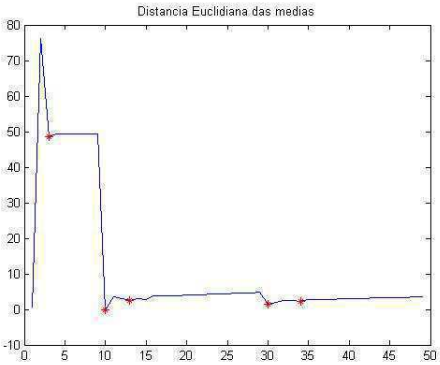
(d) Derivada da QRNS



(e) IP

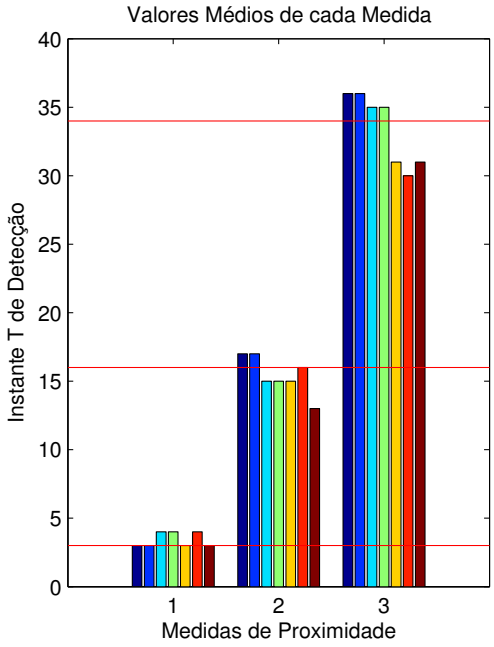


(f) Derivada do IP

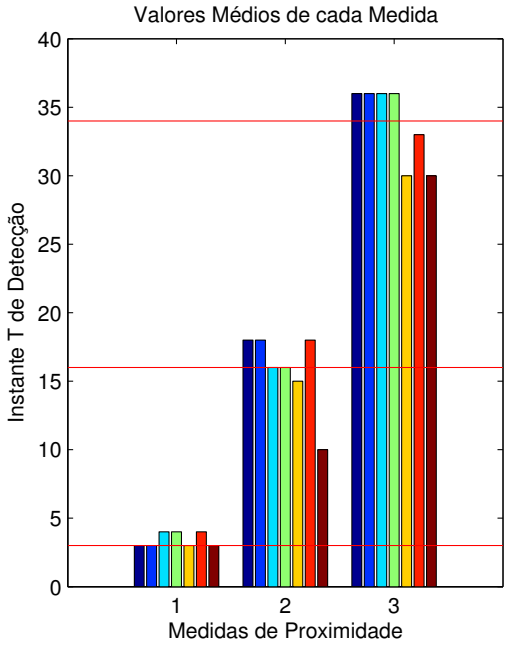


(g) Distancia Euclidiana

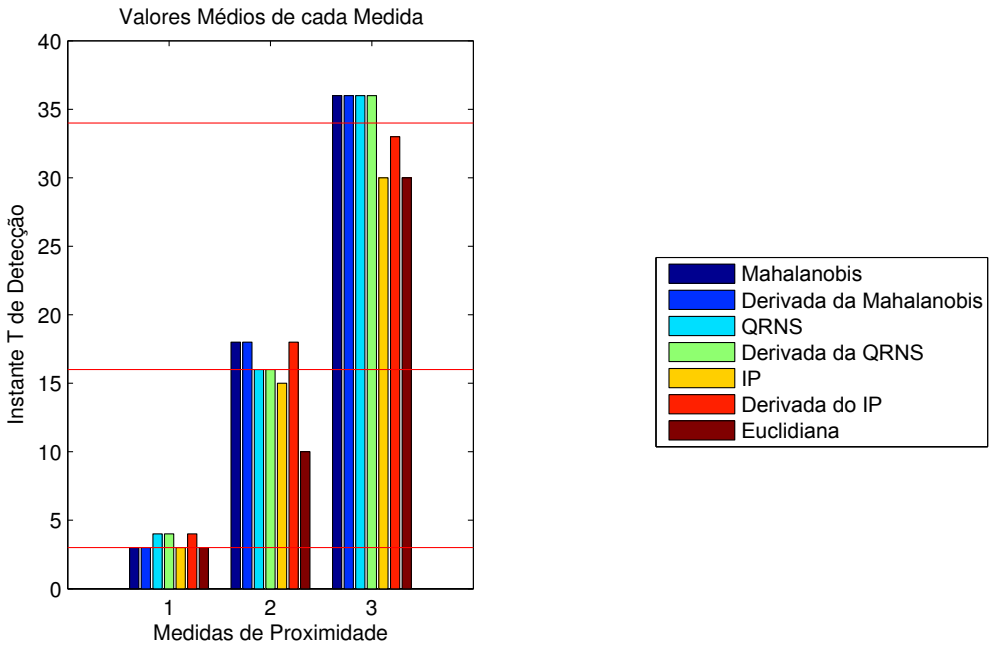
Figura 5.26: Comportamento das medidas de similaridade ao longo do tempo. Experimento 2: 3 Separações



(a) Simulação com 1.000 pontos



(b) Simulação com 10.000 pontos

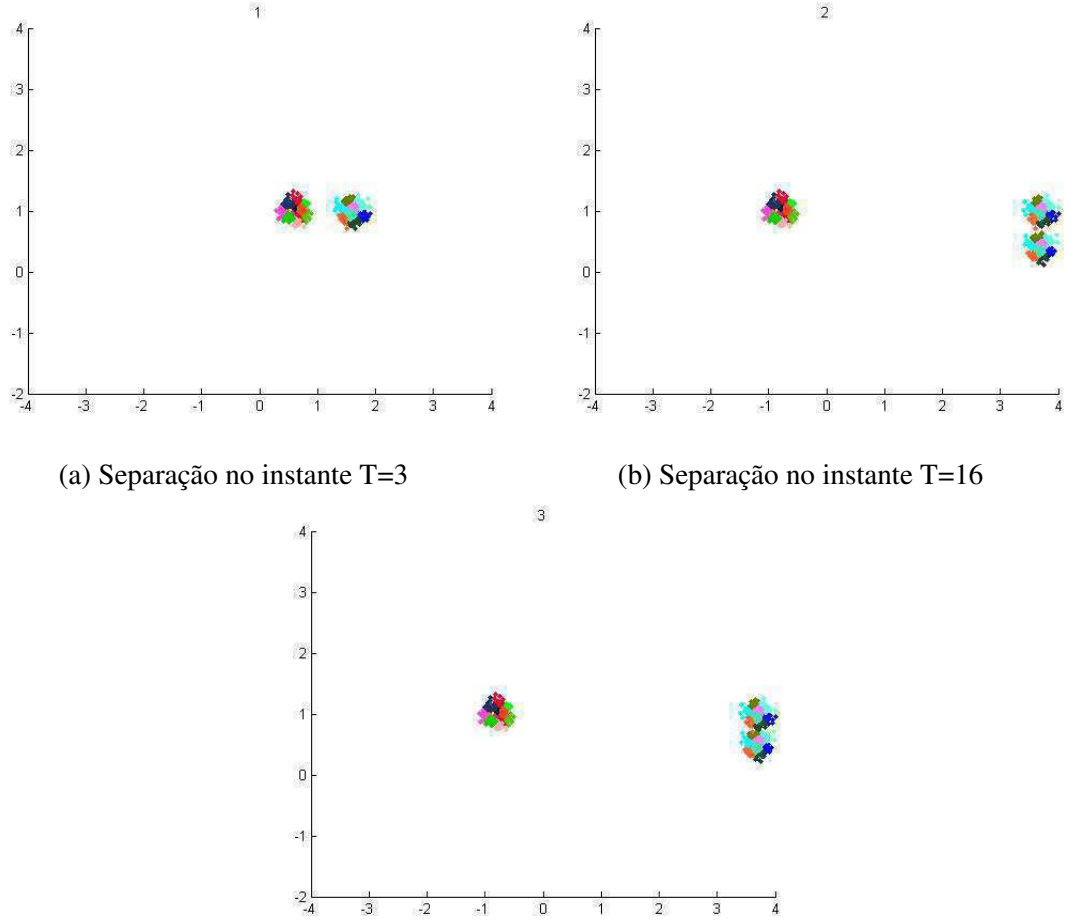


(c) Simulação com 100.000 pontos

Figura 5.27: Instante Médio das Separações - Experimento 2

Experimento 3: 2 Separações e 1 Junção entre clusters

No terceiro experimento, os dados são alterados de modo com que ocorram 2 separações e 1 junção entre os clusters : nos instantes T=3, T=16 e T=45, respectivamente. A figura 5.28 apresenta os instantes de mudanças nas estrututura dos clusters.



(a) Separação no instante T=3

(b) Separação no instante T=16

(c) Separação no instante T=45

Figura 5.28: Clusters Dinâmicos - Experimento 3: 2 Separações e 1 Junção

Foram executadas 50 iterações para cada simulação e os resultados das simulações para conjuntos com 1000, 10.000 e 100.000 pontos são apresentados nas figuras 5.29,5.30 e 5.31. Nessa figura podemos observar que novamente os algoritmos baseados na QRNS e sua derivada juntamente com o algoritmo baseado na derivada do IP foram os que obtiveram os melhores resultados.

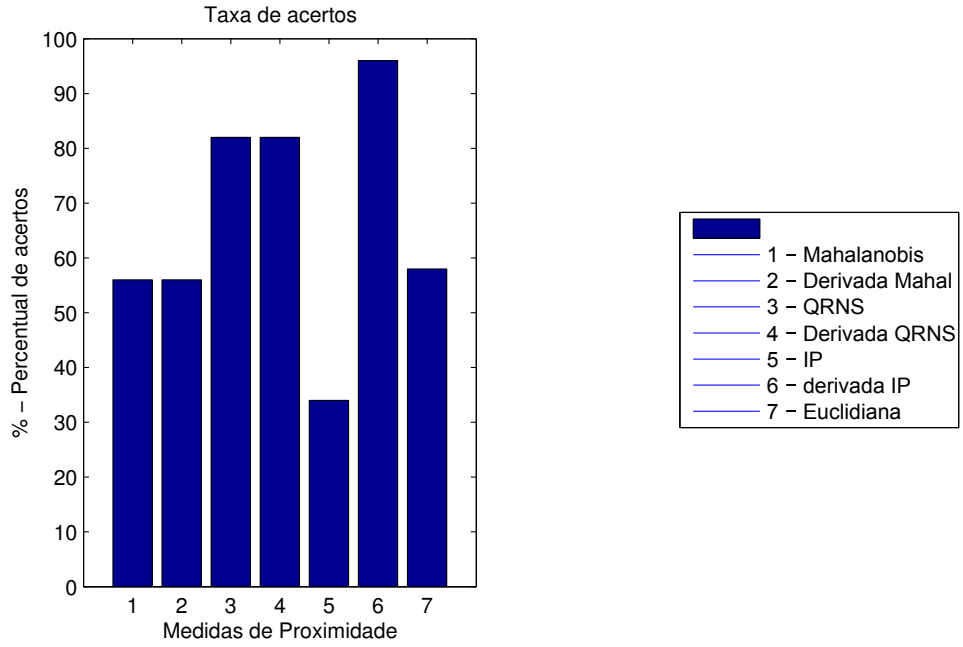


Figura 5.29: Taxa de Acertos - Experimento 3: Simulação com 1.000 pontos

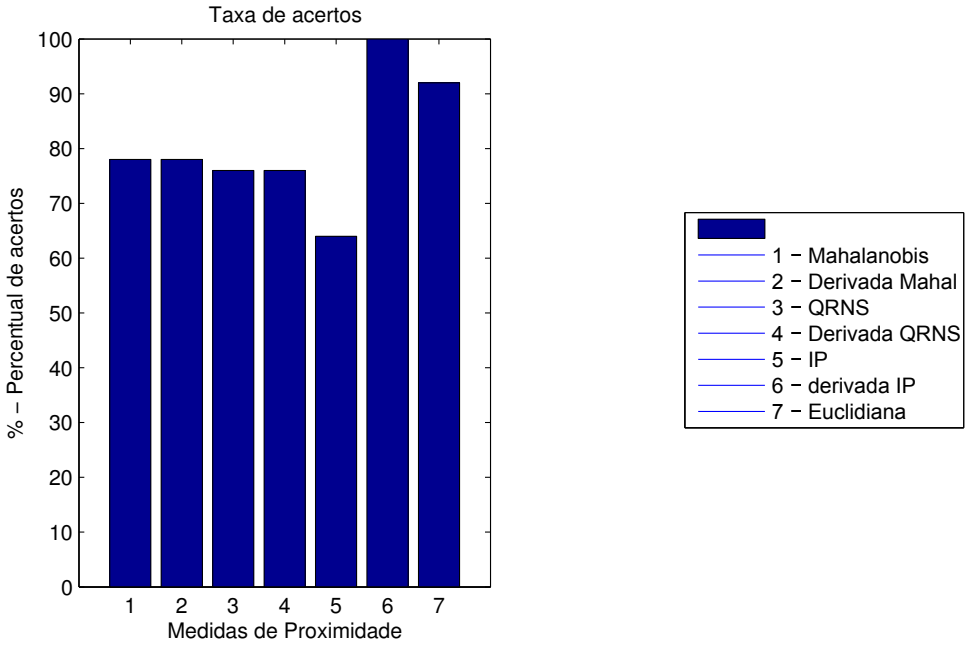


Figura 5.30: Taxa de Acertos - Experimento 3: Simulação com 10.000 pontos

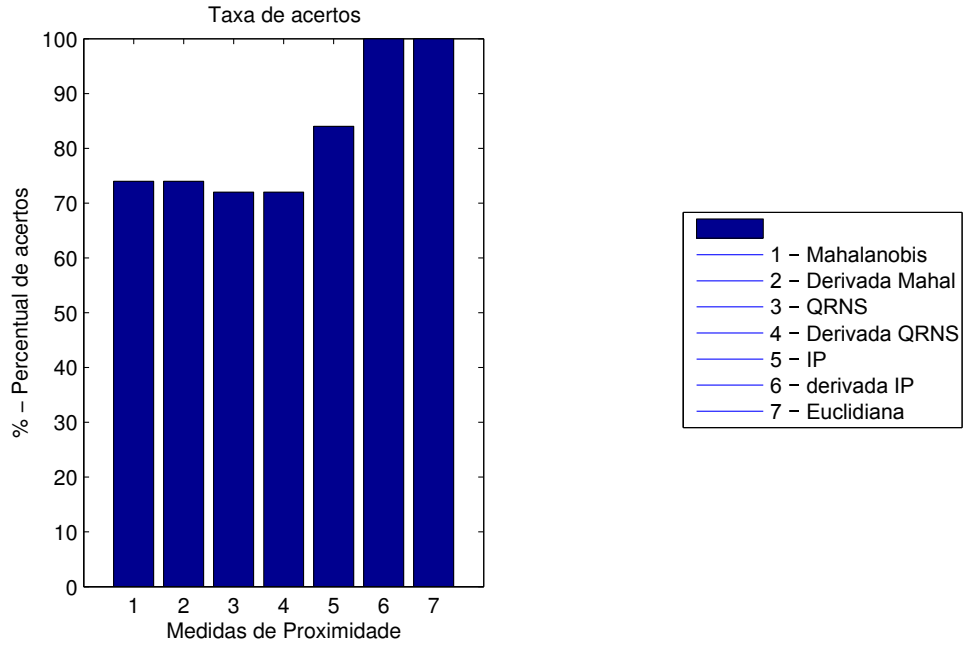
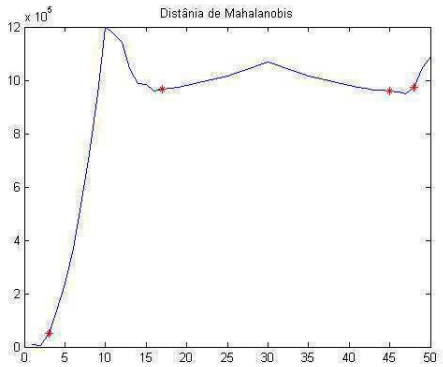
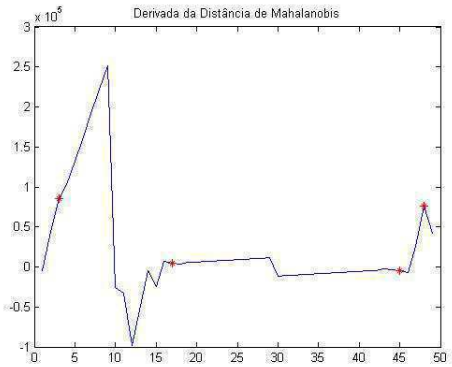


Figura 5.31: Taxa de Acertos - Experimento 3: Simulação com 100.000 pontos

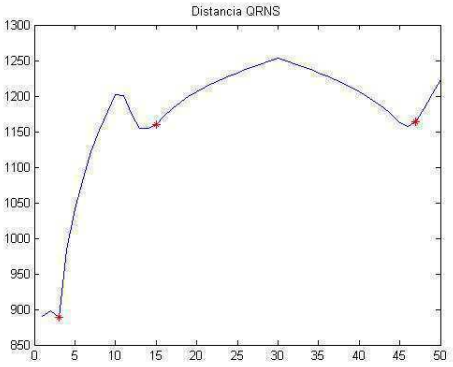
Também nota-se claramente nas figuras 5.29,5.30 e 5.31 que o desempenho das medidas melhoram quando a quantidade de pontos envolvidos no conjunto de teste aumenta. Uma exceção a isso, acontece com os algoritmos baseados na QRNS e sua derivada, que apesar de ainda apresentar um bom resultado, tem um considerável decaimento na taxa de acertos quando o conjunto de teste tem 100.000 pontos. A dificuldade dos algoritmos baseados na QRNS está em encontrar o primeiro ponto de separação, já que nos primeiros instantes de movimentação, a forma da curva apresentou variações. A figura 5.32 apresenta as curvas com o comportamento de cada medida ao longo do tempo, enquanto a figura 5.33 apresenta os instantes médios em que aconteceram as separações e a junção entre os clusters.



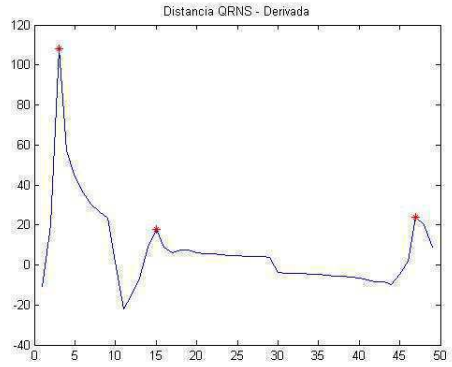
(a) Distância de Mahalanobis



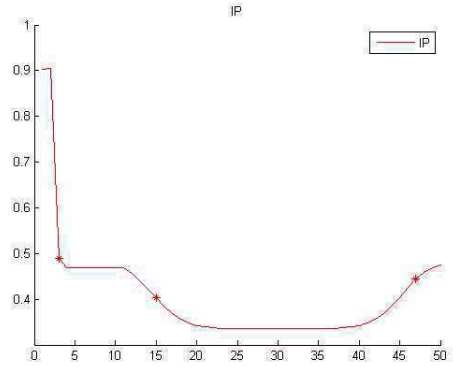
(b) Derivada da Distância de Mahalanobis



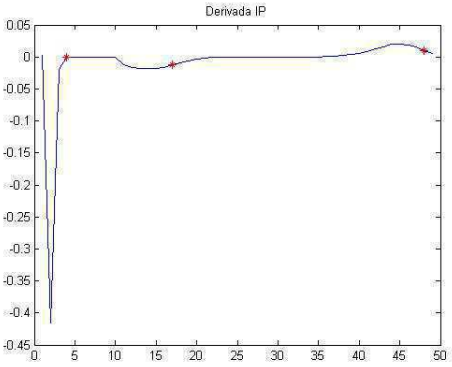
(c) QRNS



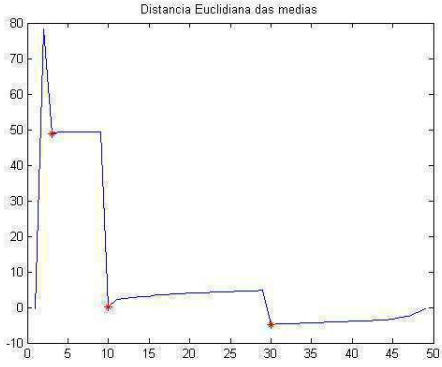
(d) Derivada da QRNS



(e) IP

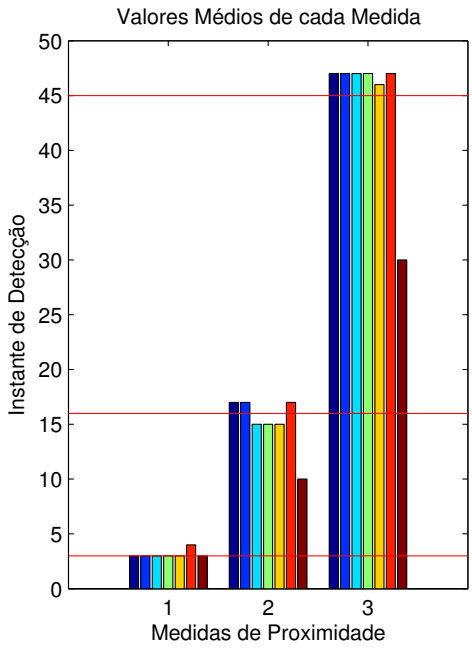


(f) Derivada do IP

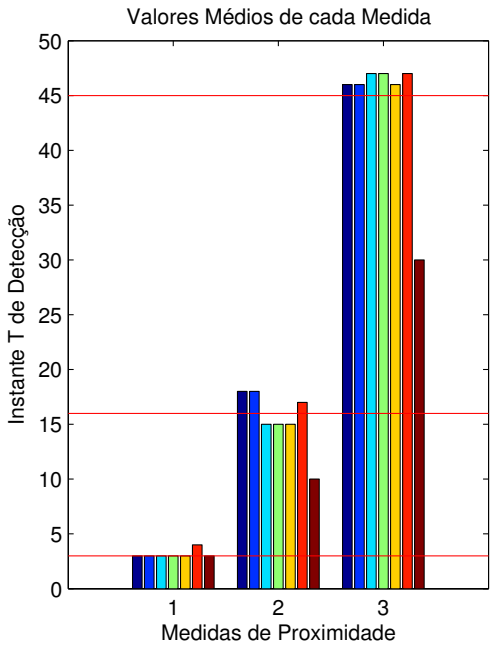


(g) Distancia Euclidiana

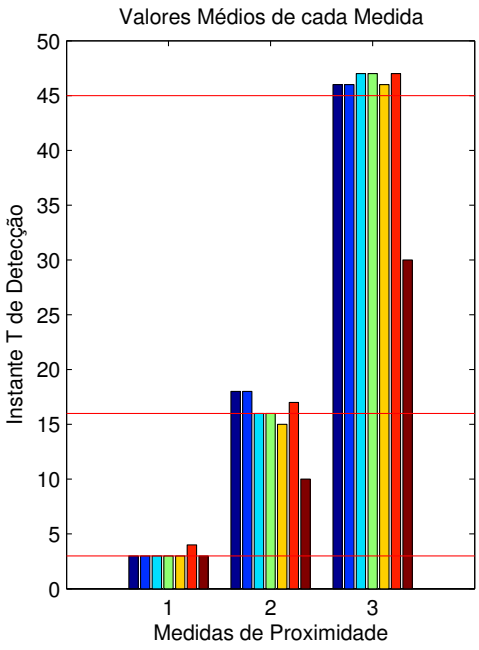
Figura 5.32: Comportamento das medidas de similaridade ao longo do tempo. Experimento 3: 2 Separações e 1 junção

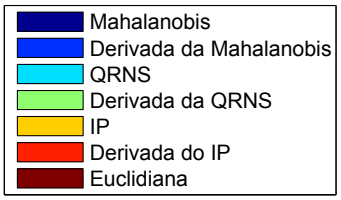


(a) Simulação com 1.000 pontos



(b) Simulação com 10.000 pontos





(c) Simulação com 100.000 pontos

Figura 5.33: Instante Médio das Separações - Experimento 3

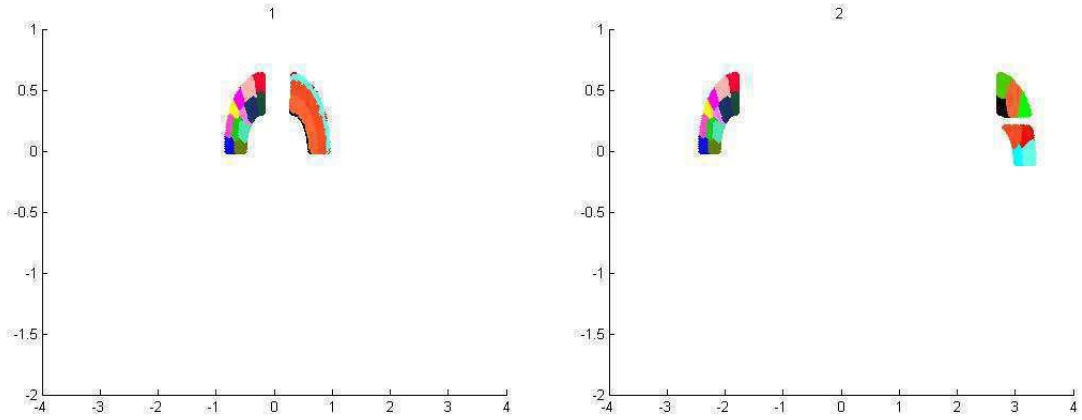
Observando a figura 5.33, notamos novamente que os pontos detectados pelo algoritmo da distância euclidiana são muito antecipados. As demais medidas apresentam bons resultados com instantes de detecção bem próximos ao esperado.

5.4.2 Toroide

Nessa segunda fase de experimentos, foram realizados testes envolvendo conjuntos que foram gerados sob a forma de uma toroide. À medida que o tempo passa, os dados são alterados de modo que os clusters formados sejam alterados em forma e quantidade.

Experimento 4: Toroide com 2 Separações

No quarto experimento, os pontos que formam a toroide são alterados de modo com que ocorram 2 separações entre os clusters : nos instantes T=2 e T=11. A figura 5.34 apresenta os instantes de mudanças nas estrututura dos clusters.



(a) Separação no instante T=2 (b) Separação no instante T=11

Figura 5.34: Clusters Dinâmicos - Experimento 4: Toroide com 2 Separações

Foram executadas 50 iterações com um conjunto de 10.000 pontos. O resultado da taxa de acerto para cada algoritmo é apresentado na figura 5.35. Nessa figura podemos observar que todos os algoritmos, com exceção do algoritmo baseado na distância euclidiana, apresentaram bons resultados. A figura 5.36 apresenta os instantes médios em que aconteceram as separações e a junção entre os clusters.

A partir da figura 5.36 podemos observar que o IP apresentou a detecção da segunda separação de forma muito retartada. Isso acontece porque o algoritmo do IP é baseado em um limiar definido pelo usuário que é dependente de cada aplicação, o que faz com que nem sempre obtenhamos bons resultados. Além a distância euclidiana mais uma vez não apresentou bons resultados, pois detectou pontos extra erroneamente.

O baixo desempenho do algoritmo baseado na distancia euclidina deve-se a detecção de pontos extras erroneamente, como pode ser observado na figura 5.37.

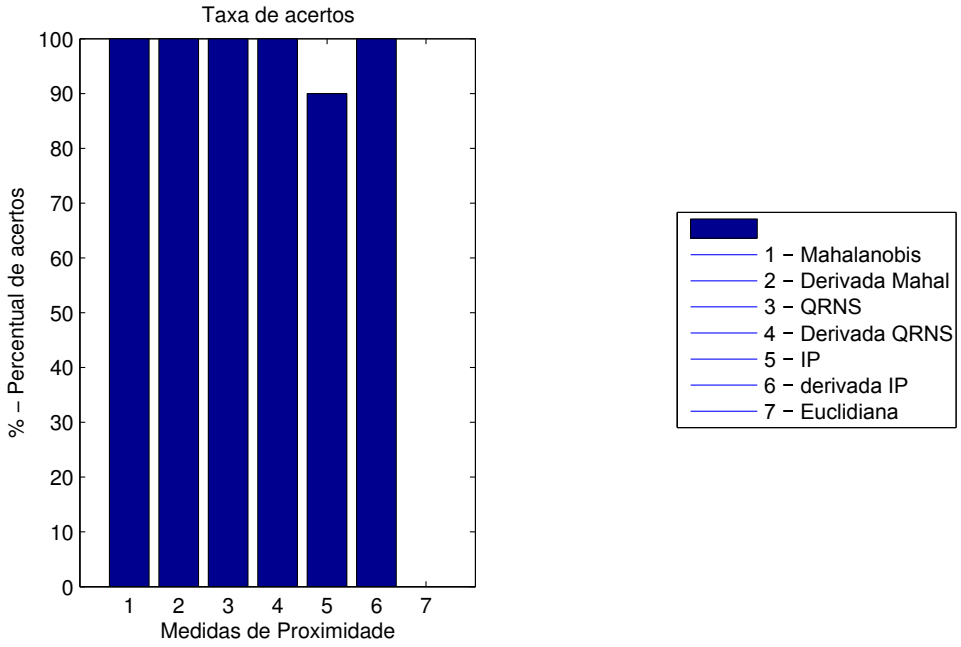


Figura 5.35: Taxa de Acertos - Experimento 4: Simulação com 10.000 pontos

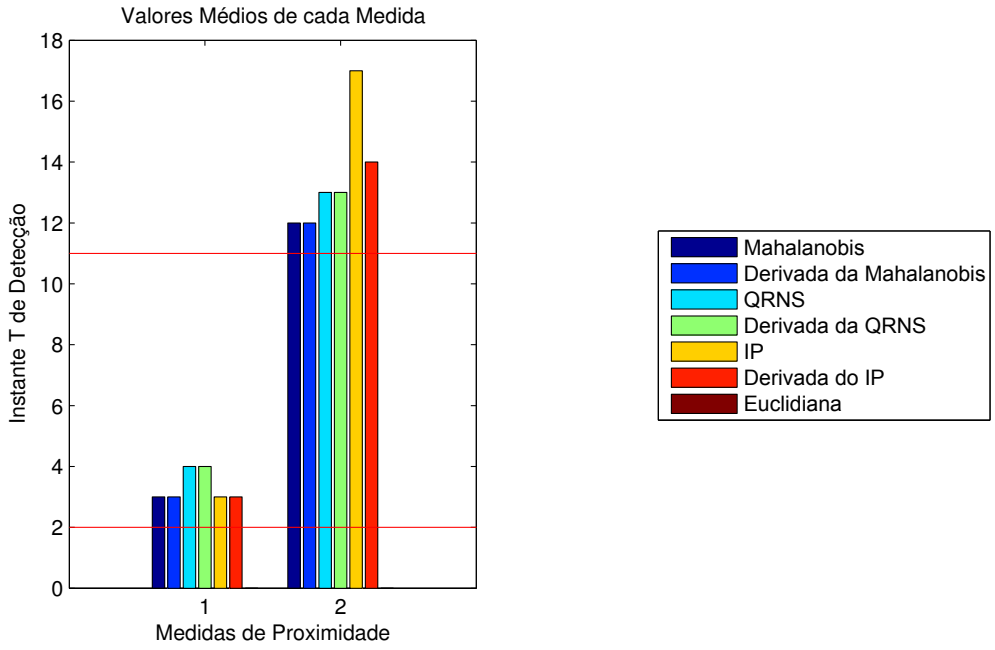
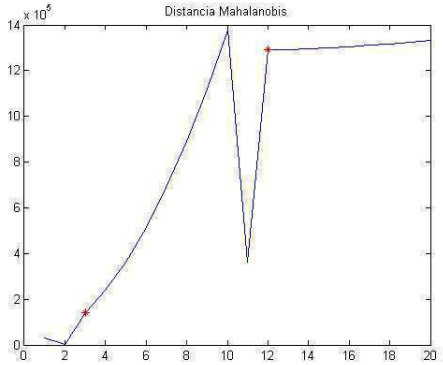
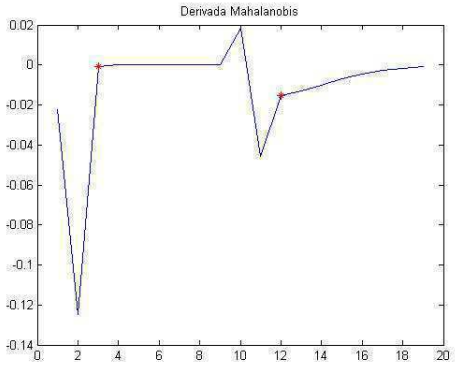


Figura 5.36: Instante Médio das Separações - Experimento 4

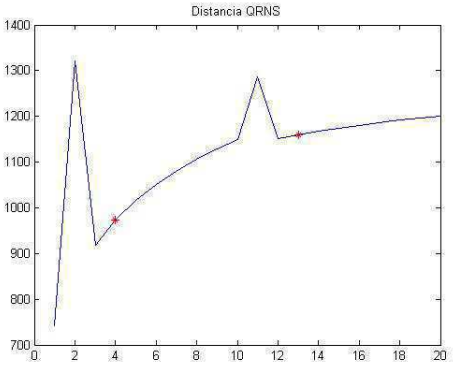
A figura 5.37 apresenta as curvas com o comportamento de cada medida ao longo do tempo.



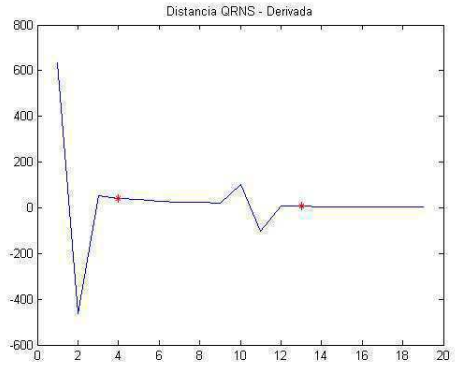
(a) Distância de Mahalanobis



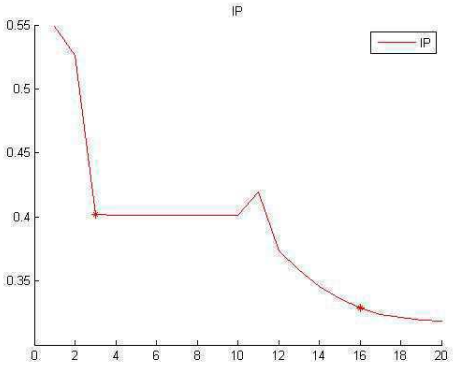
(b) Derivada da Distância de Mahalanobis



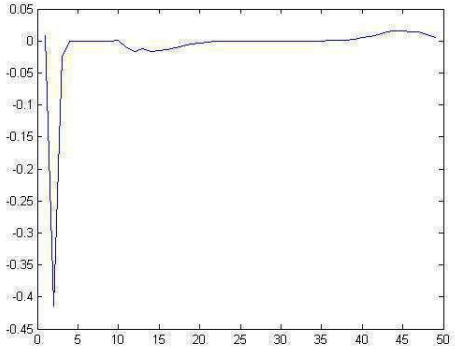
(c) QRNS



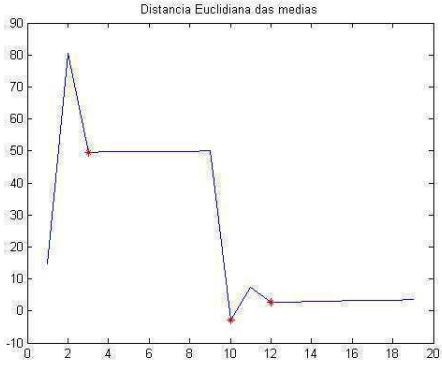
(d) Derivada da QRNS



(e) IP



(f) Derivada do IP

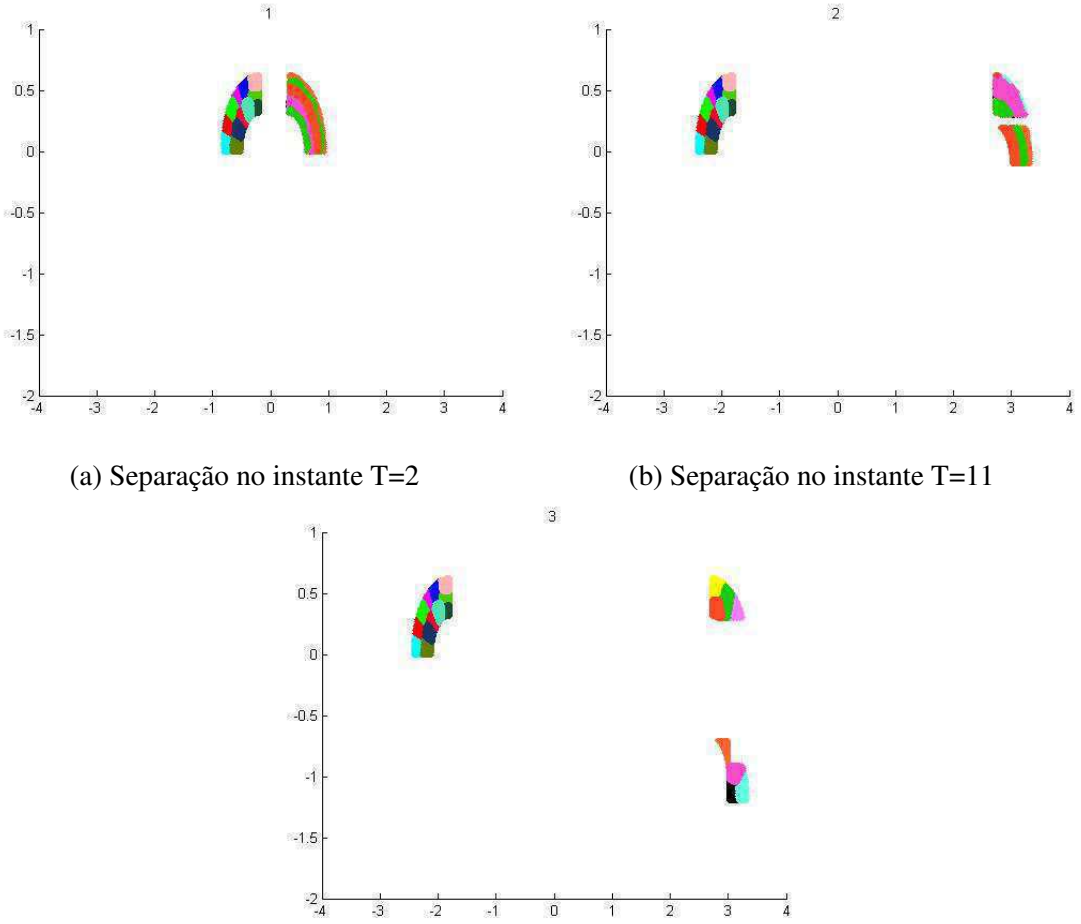


(g) Distancia Euclidiana

Figura 5.37: Comportamento das medidas de similaridade ao longo do tempo. Experimento 4: Toroide com 2 Separações

Experimento 5: Toroide com 3 Separações

No quinto experimento, os pontos que formam a toroide são alterados de modo com que ocorram 3 separações entre os clusters : nos instantes T=2, T=11 e T=22. A figura 5.38 apresenta os instantes de mudanças nas estrututura dos clusters.



(a) Separação no instante T=2

(b) Separação no instante T=11

(c) Separação no instante T=22

Figura 5.38: Clusters Dinâmicos - Experimento 5: Toroide com 3 Separações

Novamente foram executadas 50 iterações com um conjunto de 10.000 pontos. O resultado da taxa de acerto para cada algoritmo é apresentado na figura 5.39. Nessa figura podemos observar que apenas os algoritmos baseados na distanca de mahalanobis e sua derivada e na QRNS e sua derivada obtiveram bons resultados.

A figura 5.40 apresenta os instantes médios em que aconteceram as separações entre os clusters, enquanto a figura 5.41 apresenta as curvas com o comportamento de cada medida ao longo do tempo.

Note que o algoritmo da distância euclidiana novamente detecta pontos erroneamente. A dificuldade dos algoritmos baseados no IP e sua derivada está em detectar o instante da ultima separação.

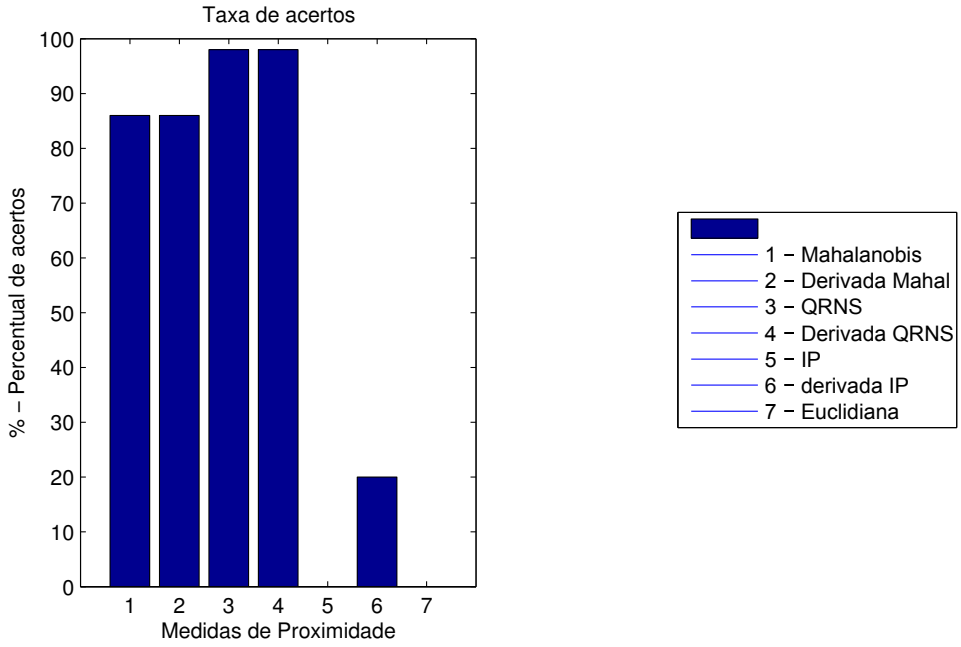


Figura 5.39: Taxa de Acertos - Experimento 5: Simulação com 10.000 pontos

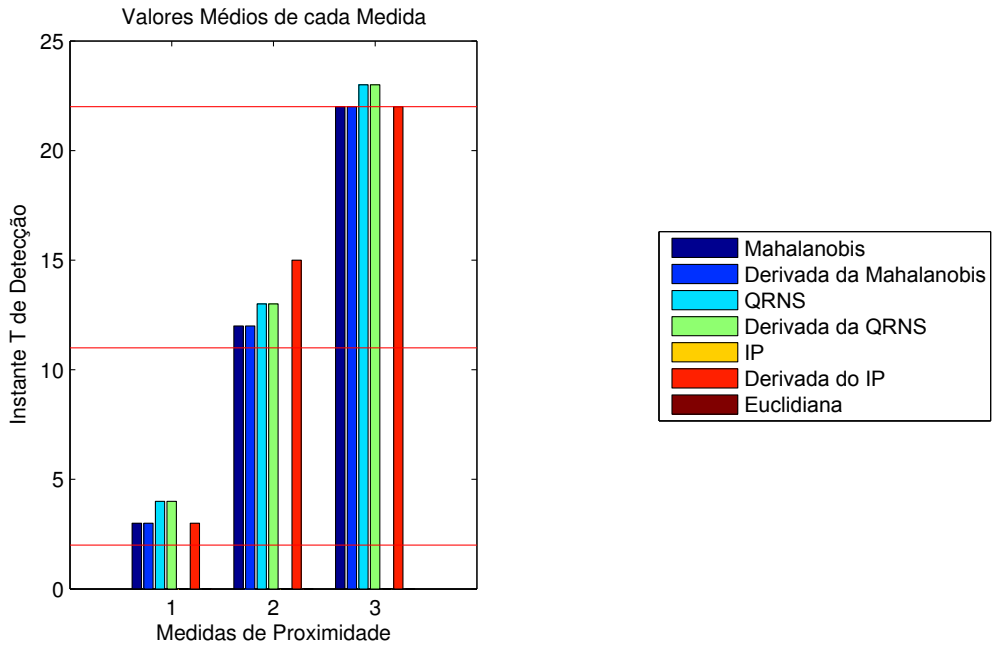
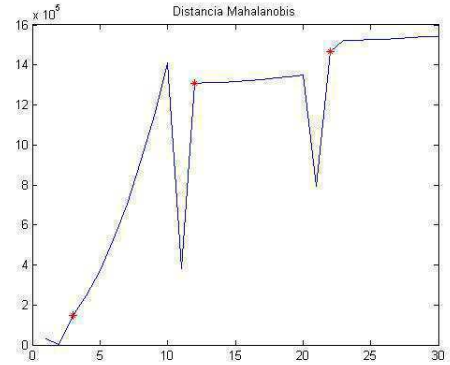
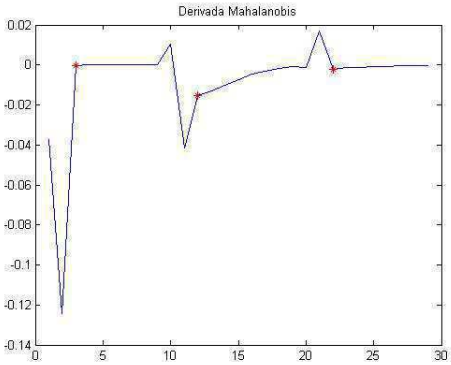


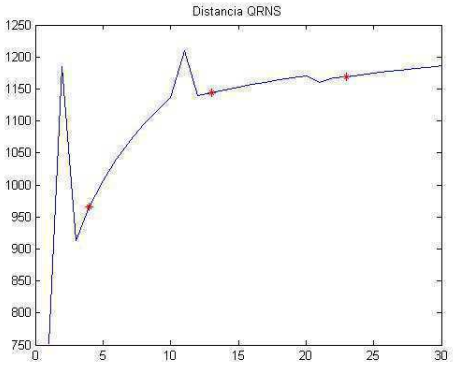
Figura 5.40: Instante Médio das Separações - Experimento 5



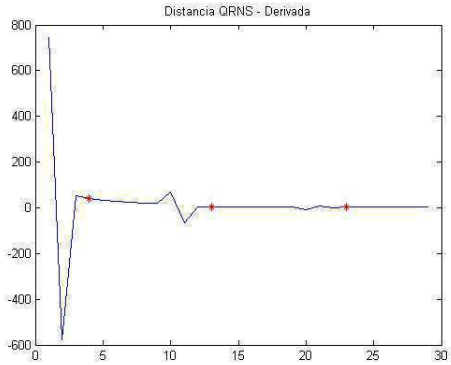
(a) Distância de Mahalanobis



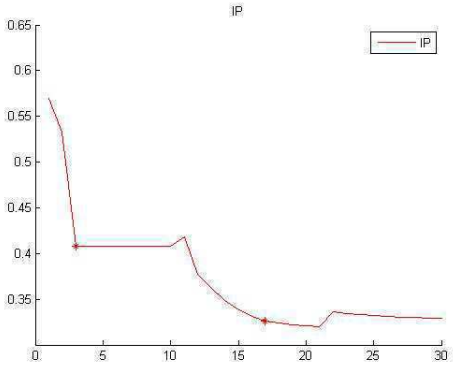
(b) Derivada da Distância de Mahalanobis



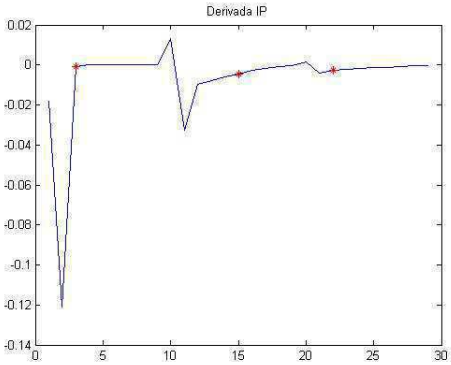
(c) QRNS



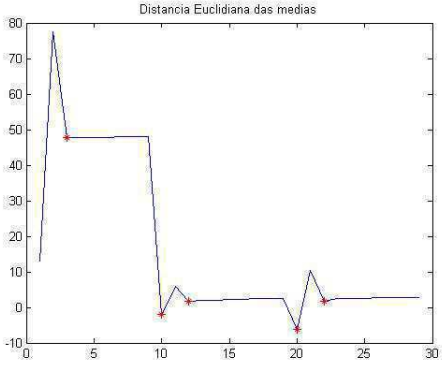
(d) Derivada da QRNS



(e) IP



(f) Derivada do IP

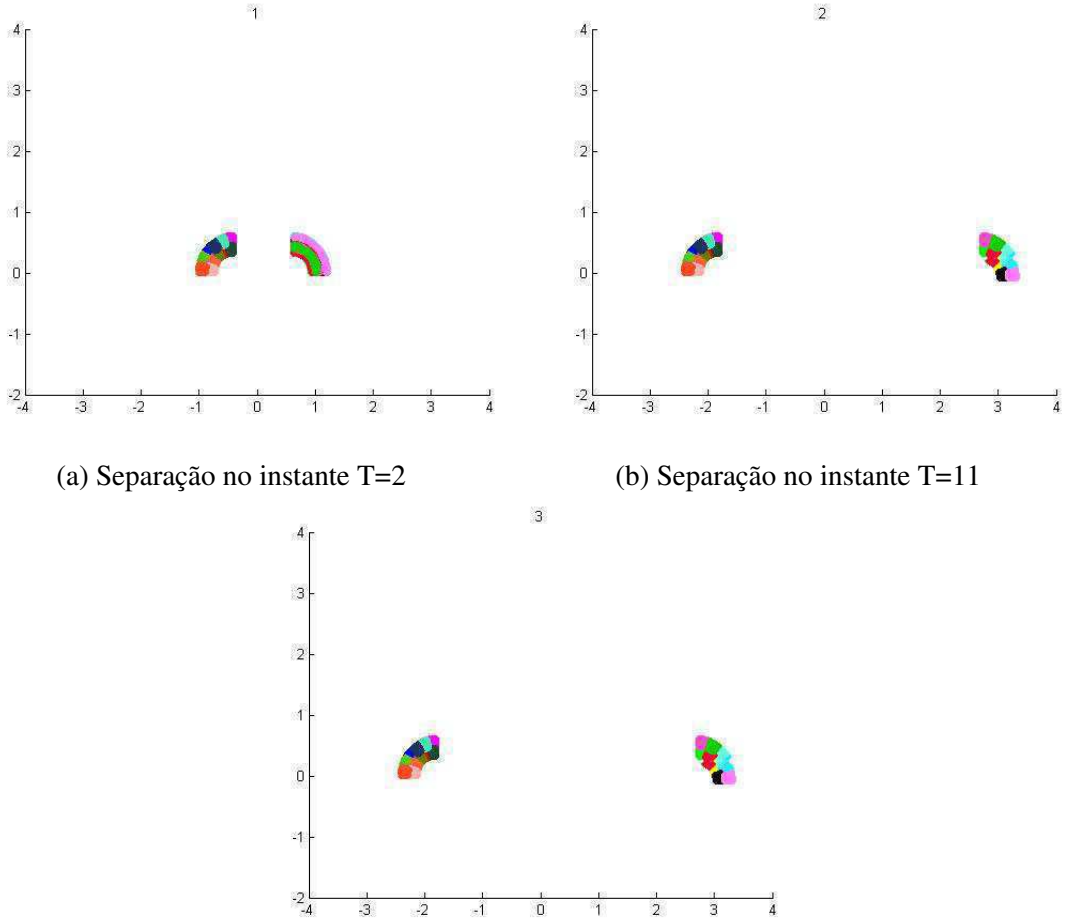


(g) Distancia Euclidiana

Figura 5.41: Comportamento das medidas de similaridade ao longo do tempo. Experimento 5: Toroide com 3 Separações

Experimento 6: Toroide com 2 Separações e 1 junção

No sexto experimento, os pontos que formam a toroide são alterados de modo com que ocorram 2 separações e 1 junção entre os clusters : nos instantes T=2, T=11e T=29. A figura 5.42 apresenta os instantes de mudanças nas estrututura dos clusters.



(a) Separação no instante T=2

(b) Separação no instante T=11

(c) Separação no instante T=29

Figura 5.42: Clusters Dinâmicos - Experimento 6: Toroide com 2 Separações e 1 junção

Novamente foram executadas 50 iterações com um conjunto de 10.000 pontos. O resultado da taxa de acerto para cada algoritmo é apresentado na figura 5.43. Nessa figura podemos observar que apenas os algoritmos baseados na QRNS e sua derivada obtiveram bons resultados.

A figura 5.44 apresenta os instantes médios em que aconteceram as separações entre os clusters, enquanto a figura 5.45 apresenta as curvas com o comportamento de cada medida ao longo do tempo.

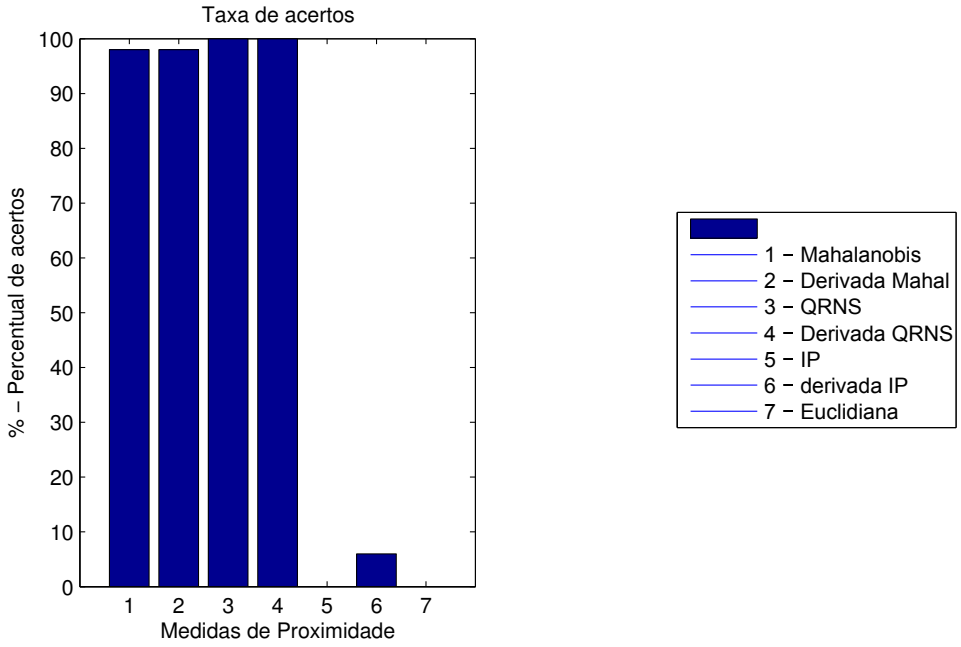


Figura 5.43: Taxa de Acertos - Experimento 6: Simulação com 10.000 pontos

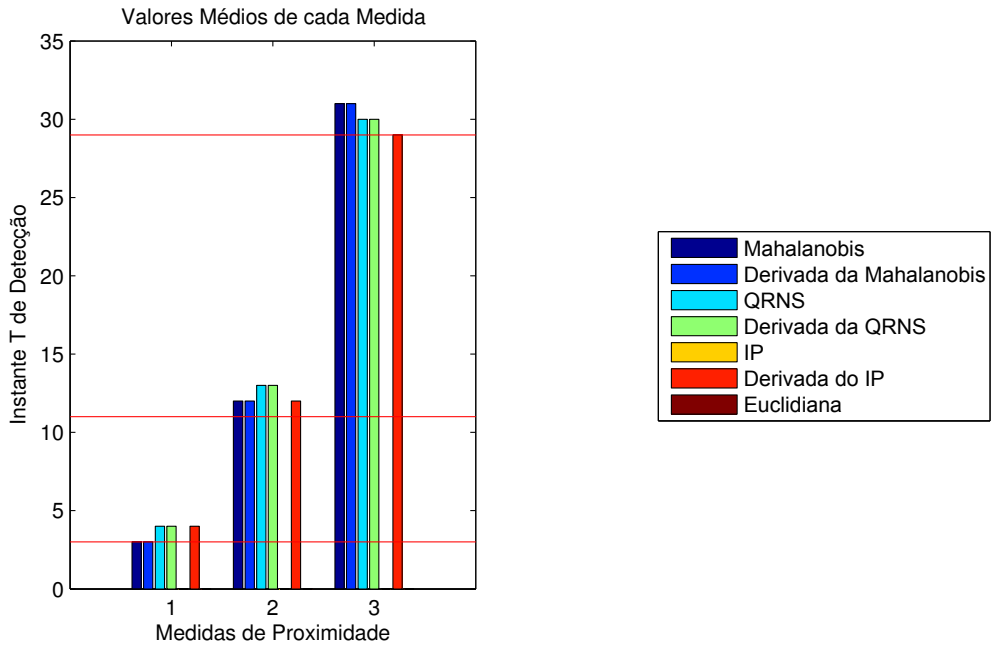
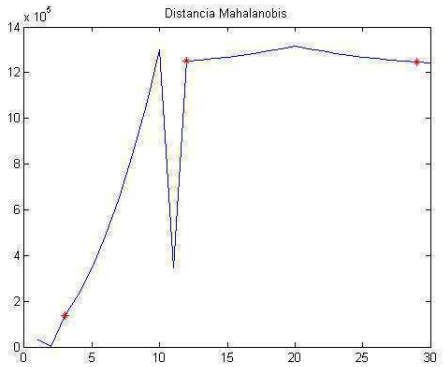
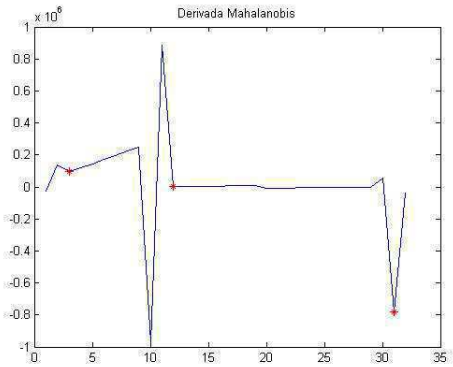


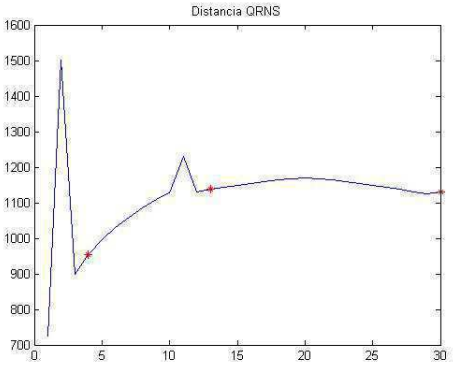
Figura 5.44: Instante Médio das Separações - Experimento 6



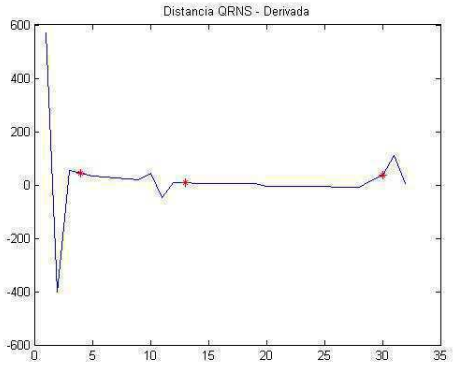
(a) Distância de Mahalanobis



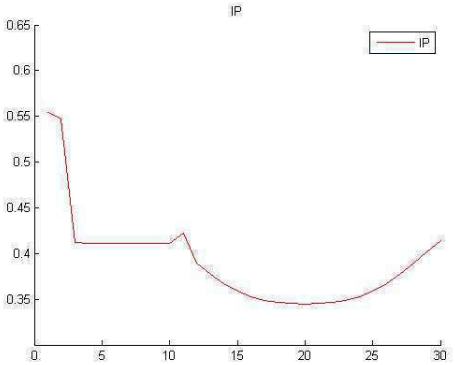
(b) Derivada da Distância de Mahalanobis



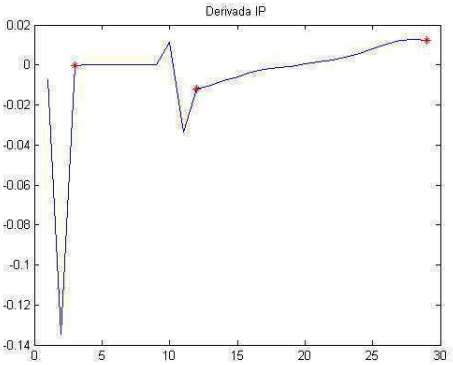
(c) QRNS



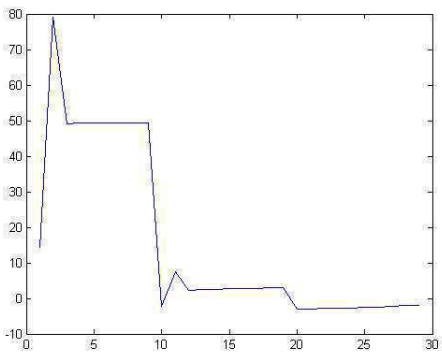
(d) Derivada da QRNS



(e) IP



(f) Derivada do IP



(g) Distancia Euclidiana

Figura 5.45: Comportamento das medidas de similaridade ao longo do tempo. Experimento 6: Toroide com 2 Separações e 1 junção

Nesse teste, o algoritmo baseado na distância euclidiana novamente detecta pontos erroneamente, enquanto que os algoritmos baseados no IP e sua derivada apresentam dificuldades em detectar o instante da ultima separação. A distância de Mahalanobis e sua derivada também tem difuldades de encontrar o instante da última separação.

Em resumo, quando analisamos o comportamento de todas as medidas nos testes o que podemos concluir é que a distância euclidiana foi a medida que apresentou os piores resultados. Em quase todos os casos, o algoritmo baseado na medida detecta pontos extras que não deveriam significar nada. Além disso, o algoritmo detecta os instantes de início de movimentação na nuvem de dados e não os instantes em que as separações e junções acontecem, o que explica as detecções antecipadas em todos os experimentos. Os algoritmos baseados no IP e sua derivada se apresentaram bem instáveis, com bons resultados em alguns experimentos e ruins em outros. Isso pode ser explicado pela dependência dos algoritmos em um limiar setado pelo usuário. O algoritmo baseado na distância de mahalanobis e sua derivada também apresentou comportamento instável entre os experimentos. A maior dificuldade nesses algoritmos foi o de encontrar pontos extras erroneamente no fim das simulações para alguns conjuntos de teste. O algoritmo baseado da medida QRNS e na sua derivada se mostrou ser o mais robusto entre os algortimos analisados. A única dificuldade encontrada com esse método está na detecção da primeira separação em alguns casos, pois algumas curvas apresentam uma certa instabilidade em sua forma nos primeiros instantes da simulação.

Capítulo 6

Conclusão

Este trabalho realizou um estudo à respeito do uso de descritores da teoria da informação dentro do contexto dos processos dinâmicos. Para isso o trabalho foi organizado em fases bem definidas, onde cada delas foi apresentada como casos de estudo distintos.

Na primeira fase realizou-se uma análise do comportamento dinâmico da informação ao longo do tempo. Os experimentos foram realizados utilizando-se vídeos como exemplos de processos dinâmicos reais. Foram realizados experimentos em 3 diferentes situações, e em todas elas, estimou-se uma grandeza estatística da teoria da informação, conhecida como potencial de informação, para todos os instantes de tempo e analisou-se o comportamento desta medida ao longo de tempo, levando-se em consideração as diferentes características existentes em cada vídeo. Além disso, foram realizadas análises por meio de uma rede neural autorregressiva com o intuito de gerar um comportamento dinâmico da informação presente nos sistemas. Esse resultado se mostrou bastante interessante, pois mostrou que a rede NAR foi capaz de identificar e medir a dinâmica da informação presente nos dados de forma adequada. Além disso, se apresenta de forma promissora, já que nos abre um leque de possibilidades de aplicações da área de teoria da informação dentro do contexto dos sistemas dinâmicos. Dentre as diversas áreas possíveis para aplicações, podemos destacar os sistemas de monitoramento por imagens, onde alertas são disparados diante de mudanças significativas no conteúdo das informações. Podemos ainda sugerir o uso da informação como uma variável adicional na modelagem dos sistemas dinâmicos por meio das técnicas tradicionais.

Na segunda fase, fez-se uso de descritores da teoria da informação na representação de processos dinâmicos por meio do uso de estados de informação em espaço de estados. Foram realizados experimentos utilizando vídeos como exemplo de um processo dinâmico. Os experimentos consistiram em obter a representação do sistema a partir do seu potencial de informação. Utilizou-se um filtro de Kalman baseado no MEI para medir ruídos nos vídeos e com isto estimar a qualidade dos mesmos com base apenas no seu potencial de informação. Em todas as simulações verificou-se que os resíduos gerados por vídeos com níveis mais altos de ruídos foram sempre maiores do que em vídeos com níveis inferiores de ruído. Foram realizados dois experimentos, onde o primeiro trata da adição de ruído gaussiano e o segundo da adição de ruído impulsivo. Em todos os casos, foram obtidos bons resultados, o que motiva uma investigação maior na técnica proposta. Vale ressaltar que o modelo estados de informação possui potencial de aplicação em diferentes áreas, tais como controle de sistemas dinâmicos, predição, clusterização dinâmica,

etc.

Finalmente, na última fase, foi investigado o comportamento de medidas baseadas da teria da informação no contexto de clusters dinâmicos, mais especificamente nas operações de merge e split entre os clusters. Foram realizados experimentos envolvendo dois conjuntos de teste distintos: um com distribuições gaussianas e outro com formato toróide. Comparando-se o desempenho das medidas para os diferentes experimentos, notou-se que os algoritmos baseados nas medidas tradicionais como distância euclidiana e de mahalanobis se apresentaram de forma bastante instável, ora apresentando bons resultados, ora maus resultados. O algoritmo baseado no IP, também se mostrou instável, e o principal motivo é sua dependência da escolha de um limiar. Porém o algoritmo baseado na medida QRNS, se mostrou bastante robusto, apresentando bons resultados em todos os experimentos realizados, independente do conjunto de testes utilizado e da quantidade de pontos envolvida. Esse resultado nos incentiva a investigarmos mais essa técnica, pois possibilita detectarmos mudanças significativas na estrutura dos clusters dinâmicos apenas pela análise do valor dessas medidas, sem a necessidade do uso de critérios geométricos. A grande vantagem em se utilizar medidas baseadas na teoria da informação ao invés de medidas tradicionais, é o fato das mesmas não fazerem suposições sobre os dados e serem calculadas diretamente das amostras. Vale ressaltar, que as todas medidas foram calculadas considerando-se apenas amostras representativas do conjunto e não todo o conjunto, o que diminui consideravelmente o custo computacinal envolvido na técnica.

6.1 Trabalhos Futuros

Como trabalhos futuros sugere-se otimizar o treinamento da rede neural autoregressiva para a predição da informação. Para isso, pretende-se melhorar a escolha dos parâmetros da rede, bem como fazer uso do critério da maximização da correntropia ao invés dos mínimos quadrados no seu treinamento.

Sugere-se ainda o aperfeiçoamento dos algoritmos para clusters dinâmicos, bem como a investigação com outras medidas de simlaridade. Além disso, a técnica necessita ser testada, com outros conjuntos de dados que envolvam distribuições e formatos diferentes.

Ainda na área de clusters dinâmicos, pretende-se explorar a técnica estudada dentro do contexto de big datas, principalmente envolvendo data streams.

6.2 Publicações

O desenvolvimento dessa tese de doutorado originou as seguintes publicações, apresentadas em ordem cronológica:

- Oliveira, A. G.; Doria Neto, A.D.; Martins, A. de M. "An Analysis of Information Dynamic Behavior". Entropy, 2017.

- Oliveira, A. G.; Martins, A. de M.; Doria Neto, A.D. "Information State: A Representation for Dynamic Processes Using Information Theory". International Joint Conference on Neural Networks (IJCNN), 2018, Rio de Janeiro.

Referências Bibliográficas

Abdallah, S. Zahraa, M. M. Gaber, B. Srinivasan & S. e Krishnaswamy (2016), 'Anynovel: detection of novel concepts in evolving data streams', *Evolving Systems* .

Aggarwal, Charu C. (2003), 'A framework for diagnosing changes in evolving data streams', *Proceedings 41st Annual Symposium on Foundations of Computer Science-Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data* .

Agrawal, R., J. Gehrke, D. Gunopulos & P. Raghavan (1998), 'Automatic subspace clustering of high dimensional data for data mining applications', *Proceedings of the 1998 ACM SIGMOD* .

Angelov, P., D. Filev & N. Kasabov (2010), *Evolving intelligent systems—methodology and applications*.

Araújo, D. (2013), Análise de Agrupamentos Com Base na Teoria da Informação : Uma Abordagem Representativa Análise de Agrupamentos Com Base na Teoria da Informação : Uma Abordagem Representativa Daniel Sabino Amorim de Araújo, Tese de doutorado.

Araújo, D., A. D. Neto & A. Martins (2013), 'Information-theoretic clustering: A representative and evolutionary approach', *Expert Systems with Applications* **40**(10), 4190–4205.

Banks, J., C. Moore, R. Vershynin, N. Verzelen & J. Xu (2018), 'Information-theoretic bounds and phase transitions in clustering, sparse pca, and submatrix localization', *IEEE Transactions on Information Theory* .

Beringer, J. & E. Hullermeier (2006), 'Online clustering of parallel data streams', *Data and Knowledge Engineering* .

Beringer, J. & E. Hullermeier (2007), 'Adaptive optimization of the number of clusters in fuzzy clustering', *Proceedings of the FUZZ-IEEE 2007* .

Bezerra, C.G. (2017), Uma Abordagem Baseada em Tipicidade e Excentricidade para Agrupamento e Classificação de Streams de Dados, Tese de doutorado.

Bifet, A., G. Holmes, R. Kirkby & R. Kirkby (2010), 'Moa: massive online analysis', *Journal of Machine Learning Research* .

Bouchachia, A. (2011), 'Evolving clustering: an asset for evolving systems', *IEEE SMC Newslett 36* .

Box, G.E.P., G. M. Jenkins & G. C. Reinsel (1994), *Time Series Analysis: Forecasting and Control*.

Campos, M.A., L.C. Rêgo & A.F. Mendonça (2016), *Métodos Probabilísticos e Estatísticos com Aplicações em Engenharias e Ciências* .

Cao, F., M. Estert & W. e Zhou A. Qian (2006), 'Density-based clustering over an evolving data stream with noise', *Proceedings of the 2006 SIAM International Conference on Data Mining* .

Carnein, M., D. Assenmacher & H. Trautmann (2017), An empirical comparison of stream clustering algorithms, *em* 'Proceedings of the Computing Frontiers Conference', ACM, pp. 361–366.

Cover, T. & J. Thomas (2006), *Elements of Information Theory*.

Declercq, A. & J. Piater (2008), 'Online learning of gaussian mixture models—a two-level approach', *Proceedings of the 3rd international conference on computer vision theory and applications* .

Ester, Martin, H. Kriegel, J. Sander & X. Xu (1996), 'A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise', *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* .

Ghesmoune, M., M. Lebbah & H. Azzag (2016), 'State-of-the-art on clustering data streams', *Big Data Analytics* .

Goil, S., H. Nagesh & A. Choudhary (1999), 'Mafia: Efficient and scalable subspace clustering for very large data sets', *Relatório técnico* .

Gokcay, E. (2000), A new Clustering Algorithm for Segmentation of Magnetic Resonance Images, Tese de doutorado.

Gokcay, E. & J.C. Principe (2002), 'Information Theoretic Clustering', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 158–171.

Gordon, A.D. (1999), *Classification*.

Guha, S., N. Mishra, R. Motwani & O'Callaghan L. (2000), 'Clustering data streams', *Proceedings 41st Annual Symposium on Foundations of Computer Science* .

Hall, P. & Y. Hicks (2005), 'A method to add gaussian mixture models', *Technical Report,University of Bath* .

Haykin, S.O (2013), *Adaptative Filter Theory*.

Hild, K.E., D. Erdogmus, K. Torkkola & J.C. Principe (2006), 'Feature extraction using Information-Theoretic Learning', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(9), 1385–1392.

Hofmann, T. & J.M Buhmann (1997), 'Pairwise data clustering by deterministic annealing', *IEEE transactions on pattern,vol 19* pp. 1–62.

Jain, A.K & R.C Dubes (1988), *Algorithms for clustering data*.

Lei, Y., J. C Bezdek, J. Chan, N. X. Vinh, S. Romano & J. Bailey (2017), 'Extending information-theoretic validity indices for fuzzy clustering', *IEEE Transactions on Fuzzy Systems* **25**(4), 1013–1018.

Liu, W., P.P. Pokharel & J.C. Principe (2006), 'Correntropy: A Localized Similarity Measure', *IEEE International Joint Conference on Neural Networks, Canada* (5), 4919–4924.

Lughofer, E. (2012), 'A dynamic split-and-merge approach for evolving cluster models', *Evolving Systems* .

Martins, A. (2005), Contribuições aos Processos de Clustering com Base em Métricas não-Euclidianas, Tese de doutorado.

Martins, A., A. Duarte, J. Dantas & Jose C. Principe (2014), 'A New Clustering Separation Measure Based on Negentropy', *Journal of Control, Automation and Electrical Systems* .

Papoulis, A. & S. U. Pillai (2002), *Probability, Random Variables ans Stochastic Processes*.

Principe, J. C., D. Xu, Q. Zhao & J. W Fisher III (2000), 'Learning from examples with information theoretic criteria', *The Journal of VLSI Signal Processing* **26**(1-2), 61–77.

Principe, Jose C (2010), *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*.

Rao, S., A. Martins & José C. Príncipe (2009), 'Mean shift: An information theoretic perspective', *Pattern Recognition Letters* (3), 222–230.

Santamaria, I., P. Pokharel & J.C. Principe (2006), 'Generalized Correlation Function: Definition,Properties, and Application to Blind Equalization', *IEEE Transactions on Signal Processing* **54**.

Şeref, O., Y. Fan, E. Borenstein & W. A. Chaovalitwongse (2018), 'Information-theoretic feature selection with discrete

-median clustering', *Annals of Operations Research* **263**(1-2), 93–118.

Silva, J. A., E.R Faria, R.C. Barros, Hruschka E.R., A.C. P. L. F. Carvalho & Gama J. (2013), 'Data stream clustering: A survey', *ACM Comput. Surv. 46(1), 13:1–13:31* .

Silva, J. de A., E. R. Hruschka & J. Gama (2017), 'An evolutionary algorithm for clustering data streams with a variable number of clusters', *Expert Systems with Applications* **67**, 228–238.

Song, M. & H. Wang (2005), 'Highly efficient incremental estimation of gaussian mixture models for online data stream clustering'.

Xie, XL. & G. Beni (1991), 'A validity measure for fuzzy clustering', *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Xu, Rui. & D.C. Wunsch II (2009), *Clustering*, Wiley.